.

# Variable selection for supervised and semi-supervised mixtures of contaminated Gaussian distributions

by

**Jorge Alfredo Sánchez Gómez**

A thesis submitted to the

College of Science and Engineering

at the University of Glasgow

for the degree of

Doctor of Philosophy

April 2024

## Declaration

I declare that all the work presented in this thesis has been done by myself under the supervision of Dr. Nema Dean and Dr. Tereza Neocleous, except where otherwise stated. This thesis represents work completed, between 2018 and 2024 in Statistics in the School of Mathematics and Statistics at the University of Glasgow.

## Abstract

Finite mixture models have the advantage of being versatile modelling tools for grouped data (Böhning, 2000; Fraley and Raftery, 1998*b*; McLachlan and Basford, 1988). This has led them to be applied in a variety of settings such as classification and clustering problems. Like any model, they have assumptions and limitations. One of the assumptions that is common, is that there are no contaminated observations present in the data (or in the classes/clusters) (Barnett et al., 1994; Becker and Gather, 1999; Bock, 2002; Gallegos and Ritter, 2009).

A popular approach to deal with this is a finite mixture model with contaminated Gaussian component distributions (Punzo and McNicholas, 2016). Each contaminated Gaussian models the data with two components, one for non-contaminated and one for contaminated data. However, a limitation of the contaminated Gaussian mixture model is that, as a complex and usually highly-parameterised model, it is not very suitable for data with a very large number of variables. The purpose of the current thesis is to extend the applicability of this model to this type of data.

In order to preserve the original variables, rather than looking at projection methods, a greedy search (Meek, 1997) approach for variable selection is customized for a mixture of contaminated Gaussian distributions in the supervised and semi-supervised learning framework. The performance of this approach in both settings is explored in both simulated and plasmode data. The criterion used to choose variables is based on classification performance. The results show that incorporating these criterion in the tailored variable selection algorithm in most cases improved the classification performance in comparison with using all variables (and often over use of the set of variables in simulations known to be the true class separating variables). Nevertheless, the performance in identifying contaminated samples was more mixed. The proposed variable selection procedure removed some variables that do not contain class information but contain contamination information. As a result, hurting the ability of the model in identifying contaminated samples specially in cases where is highly likely the presence of contamination in all variables. To summarize, the proposed variable selection algorithm seemed to perform well in both supervised and semi-supervised settings in terms of classification. However, the performance in predicting contaminated samples depends on the type of contamination and its association with class separation. There is a slight decrease in predicting contaminated samples in cases where the contamination is present in all the variables.

## Acknowledgement

This project was funded by the Ministry of Higher Education, Sccience, Technology and Innovation (SENESCYT)'s open call scholarship programme. I would like to thank the following people who helped me to make this thesis possible:

I would like to thank each of my supervisors for their patience, guidance, and support throughout the pandemic and this project. I would like to thank Dr. Nema Dean for sharing her extensive expertise of statistics with me, her suggestions, attention to detail, and questions that helped me in the preparation for my first conference. Dr. Tereza Neocleous for being a calm influence and encouraging me to keep a good balance between work and some sunshine.

This work is also dedicated to my parents. I am thankful that they have always supported me and encouraged me to pursue whatever path I have and to never fear the unknown.

Thanks also to my brother and sisters who always made sure that I was doing ok. Next, I would like to thank the friends that I have made over the years in Glasgow and beyond. Without them all, this work would never have been possible. Thanks to Calvin who has shown me some Scottish beautiful drives. Thanks to Hannah and Jessica, who were been my partners in crime going for a short walk and coffee near the office or a pint, and being supportive when I had a meltdown!. Linda, thanks for doing some cooking and introducing some Eastern European dishes when I was so busy with work.

Thanks to my Irish friends who showed me Irish lands, especially during Christmas and New Year when it was not possible for me to go home.

A lot of thanks to my flatmate and St.Silas community who were very kind and supportive during my studies.

Maddy, thanks for driving me to Wales and hosting me that time I was unwell, and was sent to rest. Vilma thanks for making me remember my past time as a swimmer and showing me how to swim in cold waters without a swimsuit, and Max for the music recommendations and the interesting conversations.

And finally, the last and most important thanks belongs to Him who defines me, "For you created my inmost being; you knit me together in my mother's womb. I praise you because I am fearfully and wonderfully made; your works are wonderful. I know that full well." Psalm 139 vs 13-14

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The notion that every object is entirely unique can sometimes be misleading. For instance, while small stones encountered during a walk near a beach may initially appear distinct, further exploration may reveal stones with shared features or attributes. "One of the most basic abilities of living creatures involves the grouping of similar objects to produce classification" (Eusebi, 2013). These similarities enable the classification of stones into groups, defined as collections of objects with common characteristics. The formation of such groups can vary depending on the criteria and characteristics used to determine an object's membership within a specific group.

The problem of allocating new observations into a group, which is a set of observations having some properties or attributes in common, has received much attention in the statistics field. The way statisticians have approached this problem has been to build a probabilistic model that can learn from observed input variables (measuring the characteristics of interest) and predict a categorical group membership variable as an output, attempting to assign new observations into the correct groups.

The process of training these models with certain amounts of data and then using them to discover hidden patterns in the data or predict the group for new observations is called learning (Bishop, 2006; El Bouchefry and de Souza, 2020; Hastie et al., 2017). This type of statistical learning involving groups falls into three types. The first is *unsupervised* learning which is when only the input variables are observed and there is no information about the output variable, i.e. there is no group information in the data. The objective in this type of learning is to discover how observations are organized or grouped.

In *supervised* learning, the information about the input variables and the output variable is known, and used during the learning process. Finally, in *semi-supervised* learning, all of the input variables are observed, but only some of the output variable is observed, so to take advantage of all the available information, unsupervised learning is combined with supervised learning (Chapelle et al., 2006).

There are different terms used to refer to the input variables included in a model depending on the field of study. In machine learning, they are known as *features* while in the statistical literature, they are known as *independent*, *explanatory* or *predictor variables* (Hastie et al., 2017). These terms will be used interchangeably in the rest of this thesis. Similarly, the output variable can be known as the *dependent*, *response* or *outcome variable*, or *group membership*. The process of using the prediction of the output variable to assign an observation to a known class, i.e. supervised learning with a categorical outcome variable, is called *classification*.

The finite mixture model framework can offer a good framework for each type of learning (Andrews and McNicholas, 2012; Andrews et al., 2011; Banfield and Raftery, 1992; Dean, 2006; Fraley and Raftery, 1998*a*, 2002; Huang and Hasegawa-Johnson, 2009). With the abundance of data available where the input variables are known but not the response variable, it is very common to find unsupervised learning problems (Albaseer et al., 2020; Singh et al., 2008). In scenarios where all input variables are observed and the unknown output variable is categorical, a probabilistic model assumes that data originate from a mixture of multivariate distributions. Each group is represented by a multivariate distribution, with parameters estimated via the expectation-maximization (EM) algorithm, as proposed in the framework of finite mixture models (McLachlan and Basford, 1988; Wolfie, 1970).

These probabilistic models developed for unsupervised learning, commonly referred to as model-based clustering (McNicholas, 2017; Peel and McLachlan, 2000), model unknown groups which are known as clusters. Each component within the mixture model is usually assumed to represent a cluster or estimated group/class within the data, leading to the common consideration of a cluster and a mixture component as equivalent entities within the finite mixture model framework.

The finite mixture modeling framework also encompasses supervised learning problems, wherein all input variables are observed, and the output variable for certain observations is known. Only the observed data with their corresponding observed output are utilized to fit the model (McNicholas, 2017). This model in the finite mixture model framework is also known as model-based discriminant analysis in the case of continuous input variables, and the groups are also called classes (McLachlan, 1992; Ripley, 2007).

In discriminant analysis, the usual distributional assumption for continuous input data is that each class is modeled with a Gaussian distribution resulting in a mixture model for the data on all classes. Typically when the covariance matrices are restricted to be the same across all classes, this is known as linear discrimimant analysis (LDA) (Fisher, 1936) and when there is no restriction it is known as quadratic discrimimant analysis (QDA) (Huberty, 1975). Fraley and Raftery (2002) extended this to allow each class to be represented by a finite mixture of Gaussian distributions, thus generalizing LDA and QDA. A semi-supervised version, which is when the labels of a small number of observations are known and a large part is unknown, of a mixture model was introduced by Chapelle et al. (2006).

The limitations of mixtures of Gaussian distributions are well known. Standard LDA and QDA are not robust enough to accommodate data that come from a non-symmetric distribution or that deviate from a Gaussian distribution. The estimates of the variance and covariance matrices when the dimension of the data is greater than the number of observations can also be unstable (Naderi, 2024).

The previously mentioned finite mixture models rest on the assumption that the observations in the groups come from the assumed parametric model (García-Escudero et al., 2010; McLachlan et al., 2006). However, in practice during data collection of certain process under study, it is likely to observe outliers (Barnett et al., 1994), which are observed values that are surprisingly far away and appear to be inconsistent with the remainder of that set of data (Barnett et al., 1994). When this assumption does not hold, parameter estimates are usually biased and the performance of the model in assigning new observations to their respective classes deteriorates. In the presence of outliers, the researcher has to take the decision whether to include or not include these unusual observations when

modeling or not.

Among the different approaches to handle outliers in clustering or classification problems are those based on mixtures and those based on trimming (C Ruwet, 2012). Mixture model approaches aim to incorporate outliers into the model (Fraley and Raftery, 2002; Peel and McLachlan, 2000) by either considering an additional mixture component - e.g. uniform component/Poisson background noise component (Banfield and Raftery, 1993; Fraley and Raftery, 1998*b*) - or using mixtures of heavy-tailed distributions (e.g. t-distributions (McLachlan et al., 2006; McLachlan and Peel, 2000)) rather than the less robust Gaussian. Conversely, trimming approaches seek to remove outliers from the dataset - e.g, TCLUST (García-Escudero et al., 2008) - and then estimate the model.

Tukey (1960), Punzo and McNicholas (2016), noting the effect of observations that deviate slightly from the reference model, proposed a finite mixture of contaminated Gaussian distributions, which is a modification to the traditional finite mixture of Gaussian distributions, to accommodate observations that deviate slightly from the reference model.

A mixture of contaminated Gaussian distributions still inherits some of the limitations of the traditional Gaussian distribution model such as: difficulty representing data that comes from non-Gaussian distributions, high-dimensional, or large datasets. This thesis seeks to address some of the limitations of this model.

The goal of this research is to extend the supervised and semi-supervised mixture of contaminated Gaussian models to deal with high-dimensional and large datasets by wrapping it in a variable selection search algorithm.

The thesis is laid out as follows. A review of the classification models available when the data is continuous and the characteristic to be predicted is categorical, a definition of contamination, and a summary of the adaptation of classification models for contaminated data is given in Chapter 2. Chapter 3 describes an implementation of a mixture of contaminated Gaussian distributions combined with a variable search algorithm to deal with high-dimensional data in a supervised learning content. Chapter 4 deals with an implementation of a mixture of contaminated Gaussian distributions combined with a variable search algorithm to deal with high-dimensional data in a semi-supervised learning content.

Finally, Chapter 5 contains a general discussion of the work presented in the preceding chapters along with some suggestions of future directions of research in this area.

# Chapter 2

# Statistical background

## 2.1 Introduction

In the previous chapter, the concepts of statistical groups and learning were introduced, along with a brief discussion of the various types of learning problems and how finite mixture models can be applied to address them. Additionally, we touched upon the limitations of mixture models in fitting observations that deviate from the assumed distribution, and we mentioned the two main strategies for dealing with such deviations. While one strategy involves removing these outliers, our focus in this work is on accommodating observations that only slightly deviate from the assumed parametric model. It's important to note that our intention is not to accommodate extreme outliers.

The objective of this chapter is to develop the definitions and concepts that will be used in the following chapters. The structure of the chapter is as follows. An overview of statistical learning and different type of learning are provided in Section 2.2, definitions for supervised learning, labelled and unlabelled data, training and testing set, and performance metrics to assess model performance are presented in Section 2.3. Unsupervised and semi-supervised learning definitions are briefly introduced in Sections 2.4 and 2.5 respectively. In Section 2.6 the finite mixture framework, the expectation maximization (EM) algorithm and the special case of Gaussian mixture models and their parameter estimation via EM algorithm are covered. The adaptation proposed in the literature to be able to model contaminated observations via a mixture of Gaussian distributions, its parameter estimation via the expectation conditional maximization (ECM) algorithm, and a brief discussion of early stopping of the ECM algorithm are in Section 2.7. The problems

of model selection and variable selection are presented in Sections 2.8 and 2.9. Finally a summary of the chapter and the next steps are given in Section 2.10.

## 2.2 Overview of statistical learning

In statistical learning, a typical scenario involves a dataset comprising $n$ observations, each associated with $p$ input variables denoted as $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, and an outcome variable $\tilde{\mathbf{z}}_i/\mathbf{z}_i$ of interest for prediction. Let us assume that the outcome variable is observed for $m$ of the $n$ observations, and the observations are reordered such that the outcome variable is observed for the first $m$ observations. In the classification setting, $\tilde{\mathbf{z}}_i$ and $\mathbf{z}_i$ are binary vectors indicating membership in one of the $G$ groups. $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ for $i = m+1, \ldots, n$ signifies the unknown group membership to predict. Similarly, $\tilde{\mathbf{z}}_i = (\tilde{z}_{i1}, \ldots, \tilde{z}_{iG})$ for $i = 1, \ldots, m$ represents the observed group membership. Hence, this data can be represented as $Y = (X, Z) = (\mathbf{x}_1, \tilde{\mathbf{z}}_1), \ldots, (\mathbf{x}_m, \tilde{\mathbf{z}}_m), \ldots, (\mathbf{x}_{m+1}, \mathbf{z}_{m+1}), \ldots, (\mathbf{x}_n, \mathbf{z}_n)$.

More generally the outcome variable $\mathbf{z}_i$ in statistical learning can take various forms: quantitative scalar, qualitative scalar, or vector. It can also be observed or unobserved. Prediction tasks are categorised into regression for quantitative observed outputs and classification for qualitative observed ones.

In the scenario where there are $G$ groups within the data, and the group membership is only available for $m$ of $n$ observations, the objective of statistical learning is to derive predictor models or rules leveraging input variables to forecast the output. This field encompasses supervised learning, unsupervised learning, and semi-supervised learning (Hastie et al., 2017; McLachlan, 1992).

Classification, a specific supervised learning instance within this scenario, involves assigning group membership labels to unlabelled observations based on a model trained on labelled data. It encompasses discriminant analysis where the continuous input variables are modeled as separate Gaussian distributions (or mixtures) for each class. When we have an unknown group membership variable this is an example of unsupervised learning known as clustering and there is a hybrid scenario when partial group information is available - semi-supervised learning. Finite mixture models can serve as a fundamental framework for

addressing learning problems involving group prediction/discovery (Bouveyron C, 2019), which will be discussed further in subsequent sections.

## 2.3 Supervised learning

In Section 2.2 the representation of data was given by $Y = (X, Z)$, where $X$ contains feature vectors and $Z$ the outcome variable of interest. Supervised learning encompasses a variety of methods that can be categorized into three groups: parametric, non-parametric and semi-parametric. Various widely used models in supervised learning, such as parametric ones: linear regression, logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), non-parametric ones: k-nearest neighbors (KNN), decision trees, etc. and semi-parametric ones: generalised additive mdoels (GAM) and support vector machines (SVM) with kernel functions (Wood, 2017) have been extensively studied (Hastie et al., 2017; James et al., 2013).

In supervised learning, it is said that the data $X$ that corresponds to the individuals is known as *labelled* when the corresponding $Z$ contains the observed labels of those observations. When $X$ is recorded but Z is not observed, it is said that data $X$ is *unlabelled* (McLachlan, 1992).

Supervised learning involves learning from examples where input data is paired with corresponding output (labelled data). During this learning process, a predictor rule is generated using the features to try to give accurate predictions of the output. The objective of supervised learning is to generalize from the provided labelled data to predict unseen or future data. The next section will present an approach to effectively utilize the available labelled data, especially in scenarios where there is an abundance of labelled data for training the model.

### 2.3.1 Training and test sets

Data serves as the foundational material for creating prediction rules, and a substantial amount of data is typically necessary for this purpose, particularly in cases involving complex models. In both statistical and machine learning contexts, it is common to divide the labelled data into two distinct sets: the training set and the test set. (In some cases, a

third set known as the validation set is also used but not during this thesis.)

The *training* data is a subset of the labelled data, consisting of $m$ observations, that is used to construct the prediction model. The *test* data comprises data distinct from those in the training set, the remaining $n - m$ observations, and is not involved in the model-building process. Its purpose is to assess the effectiveness of predictor rules; a rule that performs well in the training set may not generalize well to the test set (Hastie, Tibshirani and Friedman, 2017).

The reason for evaluating models on the the test data is that the model is likely to perform better on the data that was used to create it than on other data. The performance on the test data is an estimate of how the model will perform on future data of this kind. Models chosen on the basis of performance on the training data will often fit to patterns that may be specific only to the training data and will result in overfitting.

### 2.3.2 Performance Metrics

It is sensible to assess the performance and effectiveness of a proposed model on unseen data. This assessment is carried out through performance metrics. Various metrics have been proposed to evaluate and compare different classification models, often in order to choose the "best" one. Common metrics include accuracy, sensitivity, specificity, F1-score (Sokolova and Lapalme, 2009), area under the curve (AUC) (James et al., 2013), among others. Typically, these metrics are computed on the test set. The reason being that we expect the model to overperform on the data on which it was created (the training data), while the test data is a proxy for future, unseen data, and the metrics can better estimate future performance of the models using the test data.

If we have a single prediction on a binary/two class problem (where we refer to the two classes as positive and negative), there are four possible outcomes of the prediction:

- it is a positive case and predicted to be positive by the model - a true positive result
- it is a negative case and predicted to be positive by the model - a false positive result
- it is a positive case and predicted to be negative by the model - a false negative result
- it is a negative case and predicted to be negative by the model - a true negative result

This can be generalised for more than two classes by looking at each class in turn, setting it as the positive class and all others as the negative class.

For instance, true positives ($TP_g$) represent the number of observations predicted to be in class $g$ that actually belong to class $g$, while true negatives ($TN_g$) denote the number of observations correctly predicted not to be in class $g$. Conversely, false positives ($FP_g$) indicate the number of observations predicted to be in class $g$ that actually belong to a different class, and false negatives ($FN_g$) represent the number of observations predicted not to be in class $g$ that actually belong to class $g$.

Table 2.1 illustrates this for a binary classification case. A table like this, comparing true classification with each class being a separate column and predicted classification with each class being a separate row (or vice versa) is known as a cross classification table.

Table 2.1: Cross classification table for two categories

| | Reference | | |
|---|---|---|---|
| Predicted | Actually positive | Actually negative | Total |
| Positive | TP | FP | TP + FP |
| Negative | FN | TN | FN + TN |
| Total | TP + FN | FP + TN | Total |

We usually use elements of this type of table to define metrics that evaluate the classification models performance. Several of the most common are listed hereafter. If we have more than 2 classes, we can create such a table for each class separately with it being the positive class and the others being the negative.

In all of the following metrics, a value close to (or equal to 1) indicates a very good model. Values close to 0 indicate poor performance.

### Correct classification rate (CCR)/Accuracy

Correct classification rate, also known as accuracy, is calculated by dividing the number of correctly predicted labels by the total number of observations in the test set.

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{Total}} \tag{2.1}$$

This is also defined as:

$$Accuracy = \frac{\text{Number of correctly predicted observations}}{\text{Total}} \tag{2.2}$$

**Sensitivity/Recall**

Sensitivity, also known as recall, is the ability of a model to identify those observations with the characteristic (true positives). It is calculated by finding the proportion of actual positive observations that are predicted as positive.

$$Sensitivity = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

**Specificity**

Specificity is the ability of the model to identify those observations without the characteristic (true negatives). It is calculated by finding the proportion of actual negative observations that are predicted as negative.

$$Specificity = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{2.4}$$

**Precision**

Precision can be seen as the credibility of the model. It is the fraction of the observations correctly predicted with the characteristics (True positives) of the total observations predicted with the characteristic.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.5}$$

**F1 score**

When each class roughly contains the same number of observations the distribution is said to be balanced. However, when the number of elements in one or more of the classes is bigger than the number of elements in the rest of the classes the distribution is unbalanced. For example, if a class contains 90% of the observations, then predicting everything in that class will give a high accuracy regardless of how bad performance is for smaller classes. In a two-group scenario, if either TP or TN dominates the rest of the counts the correct classification rate would be high even if the method does a poor job predicting one of the

classes. In Equations (2.1 and 2.2) if TP is close to Total then the correct classification rate will be close to 1 regardless of TN and the opposite case is also true. This means that by using accuracy to choose a model we can end up with one that can identify correctly TP or TN but not both at the same time. It is still possible to obtain high accuracy values for poorly behaved classification models in the presence of unbalanced classes. Because of this, The F1 score is proposed for unbalanced classes (Sokolova and Lapalme, 2007):

$$\text{F1 score} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{2.6}$$

If false positives and false negatives are non-zero, the F1 score gets smaller, and if these values are zero, it will be a perfect model that has high precision and sensitivity giving an F1 value of 1. In cases where we have unbalanced data, it is useful to look at the F1 score. Additionally, if there are more than two classes these metrics can be extended by calculating them for each class and averaging them to obtain a global metric.

## 2.4 Unsupervised learning

Unsupervised learning involves the extraction of patterns in the data without information about the output variable. One example of this is clustering where the objective is an unknown grouping of the data, with the estimated groups known as clusters. Clustering methods can be categorized into three types: algorithmic, parametric, and non-parametric methods. Examples of algorithmic clustering methods include k-means and hierarchical clustering, among others (Hastie et al., 2017; James et al., 2013; Kuhn and Johnson, 2013). Model-based clustering methods are among parametric clustering methods Fraley and Raftery (2007), while density-based spatial clustering of application noise is an example of a non-parametric clustering method Sander et al. (1998). In unsupervised learning scenarios, where the output variable is unknown, it is standard practice to utilize all available unlabelled data, comprising $n$ observations, to estimate the clusters. A brief description of model-based clustering is given in Section 2.6.1.

## 2.5 Semi-supervised learning

In numerous real-world scenarios, there tends to be an abundance of unlabelled data with a scarcity of labelled data. Moreover, the process of labelling a vast amount of unlabelled

data can be costly, requiring specialized expertise, real-time experimentation, or specialized equipment operated by qualified personnel, and it consumes a significant amount of time. Consequently, semi-supervised learning methods have garnered considerable attention across various domains, as they offer a solution for scenarios where there is a large pool of unlabelled data and only a small fraction is labelled (Zhu, Gamez, Chen, Chingin and Zenobi, 2009).

Semi-supervised learning (SSL) occupies a middle ground between supervised and unsupervised learning (Chapelle et al., 2006). This arises from the fact that, alongside unlabelled data, the output variable is observed for some observations but not necessarily for all.

The aim of semi-supervised learning is to utilize both labelled and unlabelled data to improve model performance. Moreover, it has been documented to develop efficient models and enhance performance, resulting in superior outcomes compared to those achieved solely from labelled or unlabelled data, by thoroughly leveraging the information contained within the unlabelled subset (Bruce, 2001; Seok, 2014).

In semi-supervised learning, there are two distinct scenarios: inductive and transductive learning, which depend on the type of training function employed Zhu and Goldberg (2022). In inductive SSL, the objective is to predict labels for unknown examples, whereas transductive SSL aims to predict labels for the unlabeled samples within the training set. Transductive methods are generally preferred over inductive ones because they can effectively utilize the information obtained from all examples (Kostopoulos, Karlos, Kotsiantis and Ragos, 2018).

Several assumptions must be met for semi-supervised learning to yield favourable outcomes. Firstly, the unlabelled data should contain useful information. Secondly, it is assumed that the output varies gradually with distance, meaning that the high-density function is smoother in regions of high density than in those of low density. Therefore, if two points are close to each other in a high-density region, their corresponding outputs will also be similar. This is commonly referred to as the smoothness assumption. Thirdly, the assumption is made that high-dimensional data resides in a low-dimensional region.

Another assumption is that the decision boundary resides in a low-density region and does not intersect clusters of different classes (Chapelle et al., 2006).

The semi-supervised learning methods can be grouped in the following categories: graph-based methods, generative models, co-training, consistency regularization, and self-training (see, e.g. Chapelle et al. (2006); Zhu and Goldberg (2022) for further details).

### 2.5.1 Pseudo-labelled data

In semi-supervised learning, unlabelled data plays a crucial role, as the learning process involves assigning labels to this data using the model. These assigned labels, which are often treated as if they were true labels, are referred to as pseudo-labelled data in the context of semi-supervised learning. Pseudo-labelled data serves an important purpose as it artificially increases the labelled data, enabling more confident labelling of the remaining unlabelled data (Lee et al., 2013; Rizve et al., 2021). Additionally, pseudo-labelled data can be used to assess models which is known as pseudo-label validation. In pseudo-label validation, the model is first trained on a small labelled dataset. Then, this trained model is used to predict labels for the unlabelled data, generating what are known as pseudo-labels. The entire dataset is then divided into training and test sets, both of which contain labelled and pseudo-labelled data. The model is trained on the training set and its performance is evaluated by calculating performance metrics on the test set, which includes both labelled and pseudo-labelled data. In the next section one of the group methods generative models which is being used through this thesis is reviewed.

### 2.5.2 Generative models

According to Chapelle et al. (2006) generative techniques use a model family $f(\mathbf{x}, z|\boldsymbol{\theta}, \boldsymbol{\pi})$ to depict the join data distribution $f(\mathbf{x}, z)$. The concept is to represent $f(x)$ as a mixture of densities on labelled and unlabelled data, treating $z$ as an unobserved variable, and then integrating the unobserved labels to associate with the observed labels. Examples of generative models are Hidden Markov Models (HMM), Multinomial Mixture Model, and the Gaussian Mixture Model (GMM) that will be covered in Section 2.6.3 (Zhu and Goldberg, 2022).

This work focuses on generative methods, specifically in semi-supervised mixture of Gaussian distributions. In the next section, we will delve into one of the group methods

known as generative models, which plays a significant role throughout this thesis.

## 2.6 Finite mixture models

### 2.6.1 Introduction to finite mixture models

A finite mixture model is a probabilistic model that represents the presence of groups within an overall population with multivariate features, without requiring knowledge of which group the observation in the population belongs to. The mixture density is a weighted sum of individual component densities with the weights summing to 1.

So, each element $\mathbf{x}$ of the population $X$ is a random vector coming from a parametric finite mixture distribution with density

$$f(\mathbf{x}|\boldsymbol{\psi}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}|\boldsymbol{\theta}_g) \tag{2.7}$$

where $\boldsymbol{\psi} = (\pi_1, \ldots, \pi_G, \boldsymbol{\Theta})$ is the vector of parameters for the mixture model with $\pi_g > 0$ the proportion of the $g^{th}$ component or mixture weight and $\sum_{g=1}^{G} \pi_g = 1$, $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G)$ the vector of parameters of the $g^{th}$ component, and $f_g(\mathbf{x}|\boldsymbol{\theta}_g)$ the density function of the $g^{th}$ group.

The main assumption of finite mixture models is that the population is a convex combination of a finite number of densities and that these component densities are parametric densities. However, it is a common to assume that the component densities come from the same distribution family for all $g$.

Finite mixture models have been broadly used in clustering and classification problems. Moreover, in the finite mixture model framework "model-based clustering" is the term often used for clustering based on finite mixture models, and "model-based discriminant analysis" is used for supervised classification. The most popular mixture model in classification applications is the mixture of Gaussian distributions that is introduced in Section 2.6.3. The next section discusses the most common algorithm used to estimate finite mixture model parameters in the frequentist inference setting (McLachlan and Peel, 2000; McNicholas, 2017; Titterington, 1990).

### 2.6.2 Expectation-Maximization (EM) algorithm

The expectation-maximization (EM) algorithm is an iterative process comprising two main steps performed until convergence. In the expectation step (E-step), the algorithm calculates the expected value of the "complete" log likelihood function, considering both observed and unobserved data. Subsequently, in the maximization step (M-step), the algorithm maximizes this expected log-likelihood function with respect to the model parameters. The EM algorithm has been applied in a variety of problems, parameter estimation for finite mixtures being one of them (McLachlan and Basford, 1988; Redner and Walker, 1984).

Let us revisit the finite mixture model from Section 2.6.1. Let us assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are p-dimensional random vectors with density function $f(\mathbf{x}|\boldsymbol{\psi})$ (see Equation 2.7) coming from a population with $G$ groups. The general likelihood for a finite mixture model is given by

$$L(\boldsymbol{\psi}|\mathbf{X}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}_i|\boldsymbol{\theta}_g) \tag{2.8}$$

The group membership of the observations is unknown and denoted by $\mathbf{z}_1, \ldots, \mathbf{z_n}$, where $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ is a binary vector with $z_{ij} = 1$ if the $i^{th}$ observation belongs to the $g^{th}$ component and 0 otherwise. The complete data likelihood incorporating the unobserved group information is represented as

$$L_C(\boldsymbol{\psi}|\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\pi_g f_g(\mathbf{x}_i|\boldsymbol{\theta}_g)\right]^{z_{ig}} \tag{2.9}$$

#### Initial values for EM algorithm

Given the importance of selecting initial values in the EM algorithm, as it can influence the quality of parameter estimates of the speed of convergence, various methods for choosing initial values have been proposed. One method, suggested by Laird (1978), involves a grid search to set initial values for the parameters. Another approach is to initialise the parameters of the mixture distributions with random values, as proposed by McLachlan and Peel (2000). In the case of choosing random starting values McLachlan and Peel (2000) recommend applying the EM algorithm from a set of random starts as a good practice.

The EM algorithm can be considered to have converged when $l_\infty^{(r+2)} - l_\infty^{(r+1)} < \epsilon$ (provided that this difference is positive) (McNicholas, 2010) with $\epsilon > 0$ being a small constant chosen by the user, here I use 0.001.

The EM algorithm begins with an initial value $\boldsymbol{\psi}$, denoted as $\boldsymbol{\psi}^{(0)}$, and the E-step and M-step are iteratively executed until convergence is achieved. The steps of the EM algorithm for a general finite mixture model are as follows.

Initial values are set and then for the $(r+1)^{th}$ update:

**E-step**

Since the random variable $\mathbf{z}$ is a multinomial distribution, its expectation is given by

$$\hat{z}_{ig}^{(r+1)} = \frac{\hat{\pi}_g^{(r)} f_g(\mathbf{x}_i|\theta_g^{(r)})}{\sum_{h=1}^{G} \hat{\pi}_h^{(r)} f_h(\mathbf{x}_i|\theta_h^{(r)})}. \tag{2.10}$$

The predicted group memberships $z_{ig}$ are estimated by the maximum a *posteriori* probabilities (MAP)

$$\dot{z}_{ig} = MAP(\hat{z}_{ig}) = \begin{cases} 1 & \text{if } \max_h\{\hat{z}_{ih}\} = z_{ig}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

**M-step**

The estimates for the parameters of the model are obtained by maximisation of the complete log-likelihood with respect to each of the parameters in $\boldsymbol{\psi}$ (vector of parameters of the model), i.e. solving

$$\frac{\partial L_C(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = 0$$

giving an estimated set of parameters $\boldsymbol{\psi}^{(r+1)}$.

### 2.6.3 Mixture of Gaussian distributions

The general form of a Gaussian mixture distributions, commonly known as model-based clustering in the literature, for $G$ populations is given by:

$$f(\mathbf{x}_i|\boldsymbol{\psi}) = \sum_{g=1}^{G} \pi_g \frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}_g|}} \exp\left( -\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_g)\right) \tag{2.12}$$

where $f_g(\mathbf{x}_i|\boldsymbol{\theta}_g) = \frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}_g|}} \exp\left( -\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_g)\right)$ (see Equation 2.7).

In clustering problems, the group information is unknown and represented by latent variables $z_{ig}$, where $z_{ig} = 1$ if observation $i$ belongs to cluster $g$, and $z_{ig} = 0$ otherwise. The complete data-likelihood presented in Equation 2.9 and the posterior probabilities outlined in Equation 2.10 for a mixture of normal distributions can be expressed as follows:

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{n} f(\mathbf{x}_i, \mathbf{z}_i|\psi) = \prod_{i=1}^{n}\prod_{g=1}^{G} \left[\pi_g \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{z_{ig}} \tag{2.13}$$

where

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)}. \tag{2.14}$$

**M-step for Gaussian mixtures**

For the particular case of a mixture of Gaussian distributions, the maximum likelihood estimates for $\pi_g$, $\boldsymbol{\mu}_g$, and a general $\boldsymbol{\Sigma}_g$ (with no restrictions) are given by

$$\begin{aligned}
\pi_g^{(r+1)} =& \frac{n_g^{(r+1)}}{n}, \quad \text{where } n_g^{(r+1)} = \sum_{i=1}^{n} z_{ig}^{(r+1)} \\
\boldsymbol{\mu}_g^{(r+1)} =& \frac{\sum_{i=1}^{m} z_{ig}^{(r+1)} \mathbf{x}_i}{n_g^{(r+1)}} \\
\boldsymbol{\Sigma}_g^{(r+1)} =& \frac{1}{n_g^{(r+1)}} \sum_{i=1}^{m} z_{ig}^{(r+1)} (\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})^T.
\end{aligned}$$

It is evident that a typical mixture of Gaussian distributions for $G$ groups and $p$-dimensional data comprises a total of $(G-1) + Gp + Gp(p+1)/2$ parameters. As the number of features $p$ increases, so does the number of parameters needed to estimate (see Equation 2.12). This expansion can restrict the applicability of the model due to its complexity. For many cases, there may not be enough data to properly estimate the large number of parameters.

Hence, some constraints are proposed upon the covariance matrix structure to obtain a parsimonious mixture of Gaussian distributions (McNicholas, 2017). For example, three possible constraints on the covariance matrices that reduce the parameters to be estimated are $\boldsymbol{\Sigma}_g = \boldsymbol{\sigma}_g^2 \mathbf{I}_p$, $\boldsymbol{\Sigma}_g = \boldsymbol{\sigma}^2 \mathbf{I}_p$, and $\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$, where $\mathbf{I}_p$ is the identity matrix of dimension $p \times p$.

In the latter case where the covariance matrices are the same for all classes $\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$ and the maximum likelihood estimators for $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}$ are estimated using the training data with known group labels and the discrimination rule is the Bayesian rule, the classification rule obtained is linear and is equivalent to that obtained in a linear discriminant analysis (LDA) model. However, if the covariance matrix is different for each group, the classification rule obtained is quadratic and is equivalent to that obtained in the quadratic discriminant model (QDA). The constraints on the group covariance matrices proposed by Banfield and Raftery (1993) are described in more detail in Section 2.6.4.

### 2.6.4 Parsimonious mixture of Gaussian distributions

Parsimonious mixtures of Gaussian distributions are an approach to restrict elements of the covariance matrices in mixture models in an effort to reduce the number of parameters in the model (Banfield and Raftery, 1993). This is obtained through an eigenvalue decomposition of the group covariance matrices to produce a range of covariance structures that require between 1 and $Gp(p+1)/2$ parameters.

The eigenvalue decomposition applied is of the form $\boldsymbol{\Sigma_g} = \lambda_g D_g A_g D_g^T$, where $\lambda_g$ is a constant that represents the volume of the $g^{th}$ group, $D_g$ is a matrix of eigenvectors of $\boldsymbol{\Sigma}_g$ that represents the orientation of the $g^{th}$ group, and $A_g$ is a diagonal matrix whose entries are proportional to the eigenvalues of $\boldsymbol{\Sigma_g}$ accounting for the shape of the $g^{th}$ group. These constraints allow the volumes, orientation, and shapes to be equal or to vary between groups producing a range of different models (Dean, 2006). The implementations of parsimonious mixtures of Gaussian distributions available in R statistical software are listed in Table 2.2.

Table 2.2: Covariance restrictions that are available as part of the mclust package

| Model Identifier† | Volume | Shape | Orientation | Decomposition | Covariance Parameters |
|---|---|---|---|---|---|
| EII | Equal | Spherical | – | $\boldsymbol{\Sigma_g} = \lambda I$ | 1 |
| VII | Variable | Spherical | – | $\boldsymbol{\Sigma_g} = \lambda_g \mathrm{I}$ | $G$ |
| EEI | Equal | Equal | Axis aligned | $\boldsymbol{\Sigma_g} = \lambda A$ | $p$ |
| VEI | Variable | Equal | Axis aligned | $\boldsymbol{\Sigma}_g = \lambda_g A$ | $p + G - 1$ |
| EVI | Equal | Variable | Axis aligned | $\boldsymbol{\Sigma}_g = \lambda A_g$ | $pG - G + 1$ |
| VVI | Variable | Variable | Axis aligned | $\boldsymbol{\Sigma}_g = \lambda_g A_g$ | $pG$ |
| EEE | Equal | Equal | Equal | $\boldsymbol{\Sigma}_g = \lambda D A D^T$ | $p(p+1)/2$ |
| EEV | Equal | Equal | Variable | $\boldsymbol{\Sigma}_g = \lambda D_g A D_g^T$ | $Gp(p+1)/2 - (G-1)p$ |
| VEV | Variable | Equal | Variable | $\boldsymbol{\Sigma}_g = \lambda_g D_g A D_g^T$ | $Gp(p+1)/2 - (G-1)(p-1)$ |
| VVV | Variable | Variable | Variable | $\boldsymbol{\Sigma}_g = \lambda_g D_g A_g D_g^T$ | $Gp(p+1)/2$ |

† The first letter represents the volume constraint, the second letter represents the shape constraint, and the third letter represents the orientation constraint ( $E \equiv$ equal; $V \equiv$ variable; $I \equiv$ identity). The number of groups is $G$ and the data have dimension $p$.

### 2.6.5 Supervised and semi-supervised mixture of Gaussian distributions

In supervised learning the groups are typically referred to as classes because the group information is known. The dataset consists of $n$ observations, with the group information observed for all of them. A common approach is to partition this data into two subsets: one containing $m$ observations for training the model and denoted by $\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_m$ where $\tilde{\mathbf{z}}_i = (\tilde{z}_{i1}, \ldots, \tilde{z}_{iG})$ for $i = 1, \ldots, m$, and another containing $n - m$ observations denoted by $\mathbf{z}_{m+1}, \ldots, \mathbf{z}_n$ that are treated as unlabelled data. The $m$ observations in the training set are used to fit the model, and the other $n - m$ observations for testing the model. The complete data likelihood for a model-based discriminant analysis is given by:

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{m} f(\mathbf{x}_i, \mathbf{z}_i | \psi) = \prod_{i=1}^{m} \prod_{g=1}^{G} \left[ \pi_g \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right]^{\tilde{z}_{ig}} \tag{2.15}$$

In semi-supervised learning, only $m$ of the $n$ observations are labelled as belonging to one of the $G$ classes. The observations are ordered such that the first $m$ observations are labelled $\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}_{m+1}, \ldots, \mathbf{x}_n$, with their corresponding group information defined as in Section 2.2. The first $m$ observations are denoted by $\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_m$, where $\tilde{\mathbf{z}}_i = (\tilde{z}_{i1}, \ldots, \tilde{z}_{iG})$

for $i = 1, \ldots, m$, and $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ for $i = m+1, \ldots, n$, represent the unknown labels. Then, the complete data likelihood is expressed as

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{m} f(\mathbf{x}_i, \mathbf{z}_i|\psi) = \prod_{i=1}^{m}\prod_{g=1}^{G}\left[\pi_g \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{z_{ig}} \prod_{i=m+1}^{n}\prod_{g=1}^{G}\left[\pi_g \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{z_{ig}} \tag{2.16}$$

Equation 2.16 illustrates that the first factor of the likelihood pertains to labelled data (discriminant analysis), while the second factor corresponds to unlabelled data (clustering). If there were no unlabelled data, Equation 2.16 would simplify, resulting in a supervised scenario containing only the first component, with $\tilde{\mathbf{z}}_i$ representing the known group information and omitting the second component. Thus, semi-supervised learning encompasses both unsupervised and supervised components.

The assumption of the absence of outliers in each group is relaxed, and the adaptation required in the mixture of Gaussian distribution to model data with *mild* outliers is discussed in Section 2.7.1. Surijat Surijat

## 2.7 Contamination

Contamination is the process by which something is altered, becomes dirty or poisonous, due to containing either undesirable or dangerous substances. It is present in different fields and it has become a considerable concern. Pollution, for example can affect the quality of air and water, while surfaces believed to be contaminated with the virus COVID-19 helped to propagate it from one host to another. Containers can easily be contaminated with unwanted substances, and there are many ways that food contamination can occur. The more society is aware of contamination and the potential damage it can cause to physical health and social life, the greater the increase in efforts to detect and remove it (Rachman, 2004). The food authentication field aims to ensure the food fulfills its labelled description - e.g. origin (species, geographical or genetic), production methods (conventional, organic), or processing technology to mention some examples. It has grown rapidly as a result of the increase in cases of contamination reported (Danezis, 2016). This problem is considered a special case of the classification problem in the statistics field, and different contributions in the framework of finite mixture models have been proposed.

In the classification framework the term, 'contamination' is part of the broader con-

cept 'outlier' which is an observation that deviates from the type of data being scrutinized. Outliers can arise for various reasons, including errors in the collection or recording, mistakes made during the experimental process, instances that are inherent different, such as new trends. So, contaminated observations are a specific type of mild outlier since they slightly differ from the type of data of interest. Therefore, while all contaminated observations are outliers, not all outliers are contaminated observations.

Finite mixture classification models were introduced briefly in previous sections in this chapter.However, the previously proposed classification models assume that the samples that make up the classes are not contaminated. This assumption can be quite unrealistic, as there are many cases where contamination occurs, e.g. by adulterating a good to mix it with non-contaminated goods to obtain an economic profit. High-quality oils are an example of a target for adulteration. Often they are partially or completely replaced with lower-quality oils that are much cheaper but the final product is labelled as high quality (Ogrinc, 2003). It is plausible to assume that contaminated samples might have some features similar and some features quite different from the non-contaminated ones. These different features might be useful when performing analysis that leads to separating these contaminated samples from the overall sample.

Modeling these situations with the models described earlier in this chapter will affect the estimation of parameters and hence the performance of the model-based discriminant analysis. These contaminated observations are often known as outliers in the statistics field (García-Escudero et al., 2010), since they differ from the ones coming from the reference model, that in our case is assumed to be a mixture of multivariate Gaussian distributions. Statisticians classify outliers into two types, *mild* and *gross* (Ritter, 2014). Mild outliers might come from other distributions, but they can be fitted appropriately. Gross outliers are very extreme values and cannot be modelled by a probability distribution. It is often recommended to include mild outliers in the modeling process and exclude gross outliers (Punzo and McNicholas, 2016).

Punzo and McNicholas (2016) propose a contaminated mixture of Gaussian distributions to model cases where there is the presence of contaminated samples in groups with continuous data. This means that each group is modelled with two components: the first multivariate Gaussian distribution accounts for non-contaminated samples and the second

multivariate Gaussian distribution accounts for contaminated samples.

The mixture of contaminated multivariate Gaussian distributions is described in more detail in Section 2.7.1.

### 2.7.1 Contaminated mixture of Gaussian distributions

The assumption of the presence of contaminated samples in groups means that every group will have an unknown number of non-contaminated and contaminated samples. To represent this data composition, a finite mixture of two multivariate Gaussian distributions was proposed by Punzo and McNicholas (2016): one for non-contaminated $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ and another to represent the contaminated component $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_g, \eta_g\boldsymbol{\Sigma}_g)$, where the proportion of non-contaminated observations is $\alpha_g \in [0, 1)$ and the inflation of variance factor, to allow higher variability around the mean for contaminated observations, is given by $\eta_g > 1$ for each group $g$ (Punzo and McNicholas, 2016). The general form of a mixture of contaminated Gaussian distributions for $G$ groups is

$$f(\mathbf{x}_i|\boldsymbol{\psi}) = \sum_{g=1}^{G} \pi_g \left[\alpha_g \mathcal{N}\Big(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\Big) + (1-\alpha_g)\mathcal{N}\Big(\mathbf{x}_i|\boldsymbol{\mu}_g, \eta_g\boldsymbol{\Sigma}_g\Big)\right] \tag{2.17}$$

where $\boldsymbol{\psi} = \{\{\pi_g, \alpha_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \eta_g\}_{g=1}^{G}\}$ contains the parameters of the model.

The proportion of observations that belong to group $g$ is represented by $\pi_g$, $0 < \pi_g \leq 1$ and $\sum_{g=1}^{G} \pi_g = 1$, similar to the mixture models covered in Section 2.6.1.

#### *Clustering*, *classification*, and *discriminant analysis* contaminated mixture of Gaussian distributions

Model-based clustering via a mixture of contaminated Gaussian distributions was proposed by Punzo (2020, 2018) and Punzo and McNicholas (2016). It is assumed that a dataset consists of $n$ samples, which are divided into $G$ groups, with $n_g$ samples in group $g$, and $\Sigma_{g=1}^{G} n_g = n$. There are two sources of unknown information, the group membership of the $n$ samples and whether they are contaminated. The unknown group membership of the $i^{th}$ observation is represented by binary vector $\{\mathbf{z}_i\}_{i=1}^{n}$ where $\{\mathbf{z}_i\} = (z_{i1}, ..., z_{iG})$ and the unobserved label binary vector indicating non-contaminated data for the $i^{th}$ observation is denoted by $\{\boldsymbol{\nu}_i\}_{i=1}^{n}$, where $\{\boldsymbol{\nu}_i\} = (\nu_{i1}, ..., \nu_{iG})$. Here, $\nu_{ig} = 1$ if observation $i$ of group $g$ is non-contaminated and $\nu_{ig} = 0$ if observation $i$ in group $g$ is contaminated (similar to

the definition of the $z_{ig}$'s).

The complete data likelihood for an unsupervised mixture of contaminated Gaussian distributions for $G$ groups including the unobserved labels $\mathbf{z}_i$ and $\boldsymbol{\nu_i}$ is given by

$$L_C(\boldsymbol{\psi}) = \prod_{i=1}^{n}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{z_{ig}}, \quad (2.18)$$

In supervised learning, the class membership is known, and the labelled data is divided into training (first $m$ observations) and test sets (last $n-m$ observations). The class membership for labelled observations is denoted by $\{\tilde{\mathbf{z}}_i\}_{i=1}^{m}$ (defined similarly to $\mathbf{z}_i$'s except known and fixed rather than estimated). Thus, the complete data likelihood for a supervised mixture of contaminated Gaussian distributions for $G$ groups for the training data where we have observed labels can be expressed as:

$$L_C(\boldsymbol{\psi}) = \prod_{i=1}^{m}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{\tilde{z}_{ig}}, \quad (2.19)$$

In semi-supervised learning, only $m$ of the $n$ observations are labelled as belonging to one of the classes and its complete data likelihood is expressed as:

$$\begin{aligned} L_C(\boldsymbol{\psi}) = &\prod_{i=1}^{m}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{\tilde{z}_{ig}} \\ &\times\prod_{i=m+1}^{n}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{z_{ig}} \end{aligned} \quad (2.20)$$

The class membership for labelled observations is denoted by $\{\tilde{\mathbf{z}}_i\}_{i=1}^{m}$ while the class membership for the unlabelled observations is referred to as $\{\mathbf{z}_i\}_{i=m+1}^{n}$.

The parameter estimation can be computed by using the expectation conditional maximization (ECM) algorithm (Meng and Rubin, 1993). Similarly to EM the ECM algorithm is a common approach for maximum likelihood estimation when data are incomplete (Punzo, 2018). Estimation via the ECM algorithm for a supervised mixture of contaminated Gaussian distributions (see Equation 2.19) is reviewed in the following section.

**Parameter estimates for mixture of contaminated normals via expected conditional maximization (ECM) algorithm**

The ECM algorithm for the general (unsupervised) finite contaminated mixture model of Gaussian distributions given in equation 2.17 comprises three iterative steps: the E-step and multiple conditional CM-steps, which are iterated until convergence. Unlike the EM algorithm, the ECM algorithm replaces the M-step with multiple conditional maximization steps. Specifically, when estimating the parameters of a mixture of contaminated Gaussian distributions with VVV covariance (variable volume, shape, and orientation see Table 2.2) $\boldsymbol{\psi}$, the set of model parameters $\boldsymbol{\phi}$, is partitioned into $\boldsymbol{\psi_1}, \boldsymbol{\psi_2}$, where $\boldsymbol{\psi}_1 = \{\pi_g, \alpha_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}_{g=1}^{G}$ and $\boldsymbol{\psi}_2 = \{\eta_g\}_{g=1}^{G}$, and the maximization is split into two CM-steps for $\boldsymbol{\psi_1}, \boldsymbol{\psi_2}$ separately.

The complete log-likelihood for $G$ groups with data of dimension $p$ is expressed as

$$
\begin{aligned}
l_C(\boldsymbol{\psi}) = &\sum_{g=1}^{G}\sum_{i=1}^{n} \tilde{z}_{ig}\left[\ln \pi_g + \nu_{ig}\ln\left(\alpha_g \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right)\right] + \\
&\sum_{g=1}^{G}\sum_{i=1}^{n} \tilde{z}_{ig}\left[(1-\nu_{ig})\ln\left((1-\alpha_g)\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \eta_g\boldsymbol{\Sigma}_g)\right)\right] \\
= &\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln\pi_g + \tilde{z}_{ig}\nu_{ig}\ln(\alpha_g) + \tilde{z}_{ig}\nu_{ig}\ln\left(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right)\right] + \\
&\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}(1-\nu_{ig})\ln(1-\alpha_g) + \tilde{z}_{ig}(1-\nu_{ig})\ln\left(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g, \eta_g\boldsymbol{\Sigma}_g)\right)\right] \\
= &\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln\pi_g\right] + \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln(\alpha_g) + \tilde{z}_{ig}(1-\nu_{ig})\ln(1-\alpha_g)\right] + \\
&\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln\left((2\pi)^{-p/2}\left|\boldsymbol{\Sigma}_g\right|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right)\right)\right] + \\
&\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}(1-\nu_{ig})\ln\left((2\pi)^{-p/2}\left|\eta_g\boldsymbol{\Sigma}_g\right|^{-1/2}\exp\left(-\frac{1}{2\eta_g}(\mathbf{x}_i-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right)\right)\right] \\
= &\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln\pi_g\right] + \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln(\alpha_g) + \tilde{z}_{ig}(1-\nu_{ig})\ln(1-\alpha_g)\right] \\
&-\frac{p\ln(2\pi)}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\right] - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln\left|\boldsymbol{\Sigma}_g\right|\right] \\
&-\frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}(\mathbf{x}_i-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right] + \\
&-\frac{p\ln(2\pi)}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}(1-\nu_{ig})\right] - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}(1-\nu_{ig})\ln\left|\eta_g\boldsymbol{\Sigma}_g\right|\right] \\
&-\frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\frac{\tilde{z}_{ig}(1-\nu_{ig})}{\eta_g}(\mathbf{x}_i-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right]
\end{aligned}
\tag{2.21}
$$

$$
\begin{aligned}
&= \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln\pi_g\right] + \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln(\alpha_g) + \tilde{z}_{ig}(1-\nu_{ig})\ln(1-\alpha_g)\right] \\
&\qquad - \frac{p}{2}\ln(2\pi)\sum_{g=1}^{G}\sum_{i=1}^{n}\tilde{z}_{ig} - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln|\boldsymbol{\Sigma}_g|\right] \\
&- \frac{p}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\tilde{z}_{ig}(1-\nu_{ig})\ln(\eta_g) - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\left(\nu_{ig} + \frac{1-\nu_{ig}}{\eta_g}\right)(\mathbf{x}_i-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right]
\end{aligned}
$$

$$
\begin{aligned}
l_C(\boldsymbol{\psi}) &= \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln\pi_g\right] + \sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\nu_{ig}\ln(\alpha_g) + \tilde{z}_{ig}(1-\nu_{ig})\ln(l-\alpha_g)\right] \\
&\quad - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n}\left[\tilde{z}_{ig}\ln|\boldsymbol{\Sigma}_g| + p\tilde{z}_{ig}(1-\nu_{ig})\ln\eta_g + \tilde{z}_{ig}\left(\nu_{ig} + \frac{1-\nu_{ig}}{\eta_g}\right)\delta(\mathbf{x}_i,\boldsymbol{\mu}_g;\boldsymbol{\Sigma}_g)\right] \\
&\qquad - \frac{p}{2}\ln 2\pi\sum_{g=1}^{G}\sum_{i=1}^{n}\tilde{z}_{ig},
\end{aligned}
\tag{2.22}
$$

where $\delta(\mathbf{x},\boldsymbol{\mu};\boldsymbol{\Sigma}) = (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$, with $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_G)^T$, $\boldsymbol{\theta} = \{\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\eta_g\}_{g=1}^{G}$, and $S = \{\mathbf{X},\mathbf{z},\boldsymbol{\nu}\}$ the set of observed and unobserved data. We want to estimate the parameters and unobserved data from this likelihood via ECM.

### Initialization

Various strategies exist for selecting the initial values $\mathbf{z}_i^{(0)}$, $\boldsymbol{\nu}_i^{(0)}$, and $\eta_g^{(0)}$ for the ECM algorithm when applied to a mixture of Gaussian models. One approach, implemented in the **ContaminatedMixt** package (Punzo, 2018), initializes $\mathbf{z}_i^{(0)}$ by randomly assigning each observation to a group from a multinomial distribution with equal probabilities $(1/G,\ldots,1/G)$. Additionally, it sets the values of $\nu_{ig}^{(0)}$ to 1 for all $i = 1,\ldots,n$ and $g = 1,\ldots,G$, that is assigning all observations initially to be non-contaminated and initializes $\eta_g^{(0)}$ to be 1.001 for all $g = 1,\ldots,G$, a minimal amount of contamination. Other initialization strategies are discussed by Punzo and McNicholas (2016). Here, the **CNmixt** function from the **ContaminatedMixt** package is used in this work. It is clear that

a Gaussian distribution can be seen as nested in the corresponding contaminated Gaussian distribution. In particular, if a class is modelled by a Gaussian distribution, it can be obtained from their corresponding contaminated Gaussian distribution by fixing $\alpha = 1$. Then, the EM estimates of the parameters of a Gaussian distribution can be used to initialize the corresponding contaminated Gaussian distribution. The initialization strategy chosen among the different initialization strategies that supports the function **CNmixt** is **"random.post"** which randomly generates the posterior probabilities of the group membership. The suggested strategy is to initialize $\nu_{ig} = 0.99$ and $\eta_g = 1.001$ for $i = 1, \ldots, n$ and $g = 1, \ldots, G$. Then, to use these initial values and run an m-step for the first CEM iteration. Although, it is not ideal having only 1 initial random start for the posterior probabilities of the group membership since it could lead to a local maxima problem, it is convenient in terms of computation speed. The following section introduces a commonly used convergence criterion for halting the ECM algorithm.

### Convergence criterion

In this section some of the common convergence criteria used for the ECM algorithm are reviewed. The same convergence criterion presented here can be used for the EM algorithm in Section 2.6.2.

One aspect of the ECM algorithm that has garnered attention is the selection of a suitable stopping rule. Various stopping criteria have been proposed in the literature (McLachlan, 1992). Lindstrom and Bates (1988) suggest stopping the algorithm when there is no relative change in either the parameters or log-likelihood, referring to it as a lack of progress rather than convergence. This entails halting the algorithm when the difference between the last values of parameter estimates or differences between the last values of parameter estimates or differences in the last observed log-likelihood reaches a threshold defined by the researcher. Another common stopping criterion in the literature is the Aitken acceleration criterion (Aitkin and Wilson, 1980), implemented in the R package **ContaminatedMixt** and used for utilised for model fitting in this study to evaluate algorithm convergence by estimating the maximum of the log-likelihood function at each iteration. Convergence is determined by comparing the log-likelihood function's value. The Aitken acceleration at iteration (r+1) is expressed as follows:

$$a^{(r+1)} = \frac{l^{(r+2)} - l^{(r+1)}}{l^{(r+1)} - l^{(r)}}$$

where $l^{(r)}$ denotes the observed-data log-likelihood value from iteration $r$. Let $l_\infty^{(r+2)}$ denote the asymptotic estimate of the log-likelihood at iteration $r+2$. This is given by:

$$l_\infty^{(r+2)} = l^{(r+1)} + \frac{l^{(r+2)} - l^{(r+1)}}{1 - a^{(r+1)}}$$

The ECM algorithm can be considered to have converged when $l_\infty^{(r+2)} - l_\infty^{(r+1)} < \epsilon$ (provided that this difference is positive) (McNicholas, 2010) with $\epsilon > 0$ being a small constant chosen by the user, here we use 0.001.

**ECM algorithm for contaminated mixture of Gaussians - Unsupervised Case**

- Initialize $\mathbf{z}_i^{(0)}$, $\boldsymbol{\nu}_i^{(0)}$, and $\eta_g^{(0)} = 1.001$ for $i = 1, \ldots, n$ and $g = 1, .., G$, for use in the first CM-steps of the ECM algorithm to give the initial parameter estimates $\psi^{(0)}$.

Iterate the following steps for $(r+1) = 1, 2, \ldots$ until convergence.

**E-step**

- Here we compute the conditional expectations of the latent variables given the observed data and current parameter estimates.
- For each observation $i$ and group $g$, this corresponds to:

$$E_{\psi^{(r)}}(z_{ig}|\mathbf{x}_i) = z_{ig}^{(r+1)} = \frac{\pi_g^{(r)}\left(\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)}) + (1-\alpha_g^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_g^{(r)})\right)}{\sum_{h=1}^{G}\left[\pi_g^{(r)}\left(\alpha_h^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_h^{(r)}, \boldsymbol{\Sigma}_h^{(r)}) + (1-\alpha_h^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_h^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_h^{(r)})\right)\right]}, \tag{2.23}$$

and

$$E_{\psi^{(r)}}(\nu_{ig}|\mathbf{x}_i, \mathbf{z}_i) = \nu_{ig}^{(r+1)} = \frac{\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)})}{\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)}) + (1-\alpha_g^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_g^{(r)})}, \tag{2.24}$$

for $i = 1, \ldots, n$ and $g = 1 \ldots, G$, where $\mathcal{N}(x|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ denotes the multivariate Gaussian probability density function for the $g^{th}$ component with mean $\boldsymbol{\mu}_g$ and variance $\boldsymbol{\Sigma}_g$.

**CM1-step**

- For fixed $\boldsymbol{\psi}_2 = \boldsymbol{\psi}_2^{(r)}$, (i.e. $\eta_g = \eta_g^{(r)}$), the parameters in $\boldsymbol{\psi}_1$ are updated as:

$$
\begin{aligned}
m_g^{(r+1)} &= \sum_{i=1}^{n} z_{ig}^{(r+1)} \\
\pi_g^{(r+1)} &= \frac{m_g^{(r+1)}}{n} \\
\alpha_g^{(r+1)} &= \frac{\sum_{i=1}^{n} z_{ig}^{(r+1)} \nu_{ig}^{(r+1)}}{m_g^{(r+1)}} \\
S_g^{(r+1)} &= \sum_{i=1}^{n} z_{ig}^{(r+1)} \left( \nu_{ig}^{(r+1)} + \frac{1 - \nu_{ig}^{(r+1)}}{\eta_g^{(r)}} \right) \\
\boldsymbol{\mu}_g^{(r+1)} &= \frac{\sum_{i=1}^{n} z_{ig}^{(r+1)} \left( \nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}} \right) \mathbf{x}_i}{S_g^{(r+1)}} \\
\boldsymbol{\Sigma}_g^{(r+1)} &= \frac{1}{m_g^{(r+1)}} \sum_{i=1}^{n} z_{ig}^{(r+1)} \left( \nu_{ig}^{(r+1)} + \frac{1 - \nu_{ig}^{(r+1)}}{\eta_g^{(r)}} \right) (\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})^T
\end{aligned}
$$

**CM2-step**

- For fixed $\boldsymbol{\psi}_1 = \boldsymbol{\psi}_1^{(r+1)}$, the parameters in $\boldsymbol{\psi}_2$ are updated by maximizing the function (see Equation 2.22)

$$
-\frac{1}{2} \sum_{i=1}^{n} \left( p z_{ig}^{(r+1)} (1 - \nu_g^{(r+1)}) \ln \eta_g - z_{ig}^{(r+1)} \left( \nu_{ig}^{(r+1)} + \frac{1 - \nu_{ig}^{(r+1)}}{\eta_g} \right) \delta(\mathbf{x}_i, \boldsymbol{\mu}_g^{(r+1)}; \boldsymbol{\Sigma}_g^{(r+1)}) \right)
$$

with respect to $\eta_g$ under the constraint $\eta_g > 1$, for $g = 1, \ldots, G$. These are the terms involving $\eta_g$ from the complete log-likelihood. Punzo and McNicholas (2016) recommend using the **optimize()** function in the **stats** package to perform a numerical search of the maximum $\nu_g^{(r+1)}$ over the interval $(1, \eta_g^*)$ with $\eta_g^* > 1$.

**Estimating class membership and contaminated set membership**

The probabilities of the $n$ observations belonging to group $g$ for $g = 1, \ldots, G$, as well as the determination of whether an observation is contaminated or not for $n$ observations, are computed using the equations derived in the E-step. These equations utilize the final

converged parameter estimates obtained from the ECM algorithm. Each observation has a posterior probability of belonging to each group. Let us assume that there are two groups and that observation $i$ has the following posterior probabilities $z_i = (0.2, 0.8)$. There are cases where there is the need to allocate an observation to one group, so the observation is allocated to the group with the maximum a posteriori probability (MAP). In the above case $MAP(0.2, 0.8) = (0, 1)$, i.e the $i^{th}$ observation is assigned to the second group. Similarly we will assign an observation to the set of contaminated observations for group $g$ if its $v_{ig}$ is less than 0.5.

**ECM algorithm for contaminated mixture of Gaussians - Supervised Case**

In the supervised case the ECM simplifies as instead of unknown class labels $\mathbf{z}$ to be estimated, the class labels $\tilde{\mathbf{z}}$ are known for the first $m$ observations which make up the training data so these do not need to be estimated.

The ECM algorithm runs the same except the $\mathbf{z}$'s are replaced by $\tilde{\mathbf{z}}$'s which remain fixed and the summation is over $m$ training observations throughout rather than all $n$ observations. Therefore in the E-step, only Equation 2.24 is used to update the unknown contamination membership information (Equation 2.23 is not needed as these $\mathbf{z}$'s are known and fixed).

Once the ECM algorithm has converged, the contamination probabilities of all $n$ observations (both training and test data) can be calculated using the E-step Equation 2.24 with the converged parameters values in the evaluation.

The class membership probabilities for the $n - m$ test data observations are calculated using E-step Equation 2.23 with the converged parameters values in the evaluation.

**ECM algorithm for contaminated mixture of Gaussians - Semi-Supervised Case**

In the semi-supervised case, the ECM algorithm incorporates both the unknown and known class labels, denoted as $\mathbf{z}$'s and $\tilde{\mathbf{z}}$'s respectively, during the calculations in the CM1 and CM2 steps. This allow for the estimation of parameters by iteratively updating the predictions of the unknown class labels $\mathbf{z}'s$ (as shown in Equation 2.25) and the contamination labels $\boldsymbol{\nu}$'s (as depicted in Equation 2.26) during the E-step.

- Initialize $\mathbf{z}_i^{(0)}$, $\boldsymbol{\nu}_i^{(0)}$, and $\eta_g^{(0)} = 1.001$ for $i = 1, \ldots, n$ and $g = 1, .., G$, for use in the first CM-steps of the ECM algorithm to give the initial parameter estimates $\psi^{(0)}$.

Iterate the following steps for $(r+1) = 1, 2, \ldots$ until convergence.

**E-step**

- Here the conditional expectations of the latent variables given the observed data and current parameter estimates are computed.
- For each observation $i$ and group $g$, this corresponds to:

$$E_{\psi^{(r)}}(z_{ig}|\mathbf{x}_i) = z_{ig}^{(r+1)} = \frac{\pi_g^{(r)}\left(\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)}) + (1-\alpha_g^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_g^{(r)})\right)}{\sum_{h=1}^{G}\left[\pi_g^{(r)}\left(\alpha_h^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_h^{(r)}, \boldsymbol{\Sigma}_h^{(r)}) + (1-\alpha_h^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_h^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_h^{(r)})\right)\right]}, \tag{2.25}$$

and

$$E_{\psi^{(r)}}(\nu_{ig}|\mathbf{x}_i, \mathbf{z}_i) = \nu_{ig}^{(r+1)} = \frac{\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)})}{\alpha_g^{(r)}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \boldsymbol{\Sigma}_g^{(r)}) + (1-\alpha_g^{(r)})\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_g^{(r)}, \eta_g^{(r)}\boldsymbol{\Sigma}_g^{(r)})}, \tag{2.26}$$

**CM1-step**

- For fixed $\boldsymbol{\psi}_2 = \boldsymbol{\psi}_2^{(r)}$, (i.e. $\eta_g = \eta_g^{(r)}$), the parameters in $\boldsymbol{\psi}_1$ are updated as:

$$\begin{aligned}
m_g^{(r+1)} &= \sum_{i=1}^{m}\tilde{z}_{ig} + \sum_{i=m+1}^{n} z_{ig}^{(r+1)} \\
\pi_g^{(r+1)} &= \frac{m_g^{(r+1)}}{n} \\
\alpha_g^{(r+1)} &= \frac{1}{m_g^{(r+1)}}\left(\sum_{i=1}^{m}\tilde{z}_{ig}\nu_{ig}^{(r+1)} + \sum_{i=m+1}^{n} z_{ig}^{(r+1)}\nu_{ig}^{(r+1)}\right) \\
S_g^{(r+1)} &= \sum_{i=1}^{m}\tilde{z}_{ig}\Big(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\Big) + \sum_{i=m+1}^{n} z_{ig}^{(r+1)}\Big(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\Big) \\
\boldsymbol{\mu}_g^{(r+1)} &= \frac{1}{S_g^{(r+1)}}\left(\sum_{i=1}^{m}\tilde{z}_{ig}\Big(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\Big)\mathbf{x}_i + \sum_{i=m+1}^{n} z_{ig}^{(r+1)}\Big(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\Big)\mathbf{x}_i\right) \\
\boldsymbol{\Sigma}_g^{(r+1)} &= \frac{1}{m_g^{(r+1)}}\sum_{i=1}^{m}\tilde{z}_{ig}\left(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\right)(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})^T + \\
&\quad \frac{1}{m_g^{(r+1)}}\sum_{i=m+1}^{n} z_{ig}^{(r+1)}\left(\nu_{ig}^{(r+1)} + \frac{1-\nu_{ig}^{(r+1)}}{\eta_g^{(r)}}\right)(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})(\mathbf{x}_i - \boldsymbol{\mu}_g^{(r+1)})^T
\end{aligned}$$

**CM2-step**

- For fixed $\boldsymbol{\psi}_1 = \boldsymbol{\psi}_1^{(r+1)}$, the parameters in $\boldsymbol{\psi}_2$ are updated by maximizing the function (see Equation 2.22)

$$
\begin{aligned}
&-\frac{p}{2}\sum_{i=1}^{m}\tilde{z}_{ig}(1-\nu_g^{(r+1)})\ln\eta_g - \frac{p}{2}\sum_{i=m+1}^{n} z_{ig}^{(r+1)}(1-\nu_g^{(r+1)})\ln\eta_g + \\
&-\frac{1}{2}\sum_{i=1}^{m}\tilde{z}_{ig}\left(\frac{1-\nu_{ig}^{(r+1)}}{\eta_g}\right)\delta(\mathbf{x}_i,\boldsymbol{\mu}_g^{(r+1)};\boldsymbol{\Sigma}_g^{(r+1)}) \\
&-\frac{1}{2}\sum_{i=m+1}^{n} z_{ig}^{(r+1)}\left(\frac{1-\nu_{ig}^{(r+1)}}{\eta_g}\right)\delta(\mathbf{x}_i,\boldsymbol{\mu}_g^{(r+1)};\boldsymbol{\Sigma}_g^{(r+1)}).
\end{aligned}
$$

### 2.7.2 Evaluation of an early stop in the (ECM) algorithm for a contaminated mixture of Gaussian distributions

As part of this thesis, many models will be fit in succession using the ECM algorithm which can take a substantial amount of time. For example, if an appropriate variance matrix restriction is required to be selected, several models need to be fitted to be compared. Meng (1994) pointed out that the ECM can be seen as a composition of two linear iterations: expected maximization EM and conditional maximization CM. Therefore, the rate of convergence of the ECM can be expressed in terms of the rate of convergence of the EM and CM, and the ECM has slowest convergence in the cases when EM and CM share a slowest component. The added simplicity and stability of the ECM algorithm over the EM comes with the price of a slower convergence rate Sexton and Swensen (2000). It is widely acknowledged within the field that the EM algorithm tends to yield substantial improvements in log-likelihood during its initial iterations. Given that the ECM algorithm represents a derivative of the EM algorithm, the prospect of prematurely halting the ECM algorithm without compromising the model's performance is explored in this section. Here, the stopping criteria based on relative change of the estimate or log-likelihood is not used in this work because of its slow convergence and the EM algorithm is stopped after ten iterations (Biernacki and Govaert, 1999; Lindsay, 1995; McLachlan and Peel, 2000). This could offer important computational speedups when fitting multiple models, but it is only appealing to do if it does not compromise model fit too much. Specifically, this line of inquiry pertains to the utilization of parameters estimated during an early stoppage of the ECM algorithm to conduct an E-step, thereby enabling predictions for observations

within the test set.

To explore an early stop of the ECM algorithm, data from one contaminated Gaussian distribution with 5% of contamination is simulated. Although the group membership and contamination information is known for observations in the test set, they are treated as unknown to measure the performance of the model identifying contamination observations in new data. The number of observations generated was 700, with half of them being allocated to the training set and half to the test set with a contamination inflation factor of 30. The parameters of the contaminated Gaussian distributions considered are thus:

Single normal with 5% of contaminated samples

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \alpha = 0.95, \eta = 30, n = 700, m = 350$$

The dataset contains only one class where around 5% of its observations are expected to be contaminated. Contamination labels are known for all 700 observations. The model was fitted with the training set composed of 350 observations, leaving the remaining as a test set to assess the model.

The **CNmixt** function from the **ContaminatedMixt** R package was employed to fit the model. Subsequently, after estimating the parameters via maximum likelihood, a custom E-step, was added to calculate predictions of contaminated membership for the test set.

Estimates of the parameters at different steps of the ECM algorithm are shown in Table 2.3. The estimates at different ECM steps reveal that parameter estimates seemed to stabilise in this case after the $10^{th}$ step. In addition to this, the parameter estimates at the $10^{th}$ iteration are pretty close to the true values and the converged values for all parameters but $\eta$. The row $\star Actual$ contains the parameter estimates obtained after running a m-step with the observed contamination labels as initial values for $v's$ and $\mu_0$ and $\Sigma_0$ estimated from the train set.

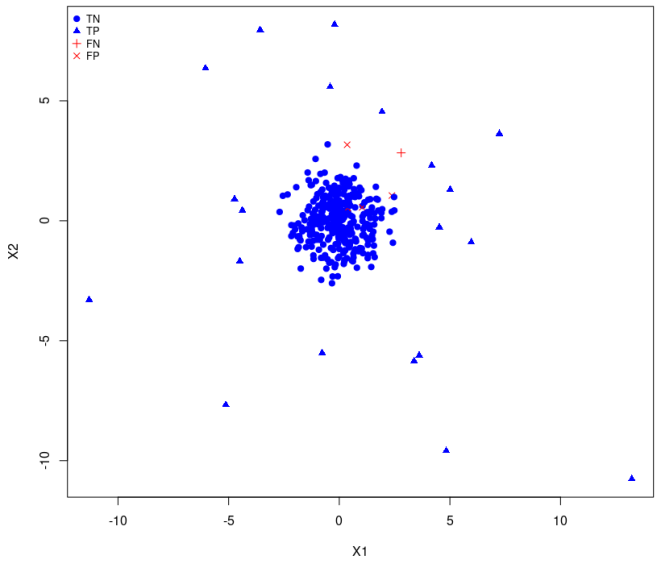
Table 2.3: Parameter estimates at selected ECM steps for a contaminated Gaussian distribution with 5% of contaminated samples

| | **Estimates** | | | | |
|---|---|---|---|---|---|
| Step | $\mu$ | $\Sigma$ | $\alpha$ | $\eta$ | complete log-likelihood |
| $3^{rd}$ | $\begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 2.75 & 0.00 \\ 0.00 & 2.75 \end{pmatrix}$ | 0.99 | 1.02 | -1367.27 |
| $5^{th}$ | $\begin{pmatrix} 0.04 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 2.52 & 0.00 \\ 0.00 & 2.52 \end{pmatrix}$ | 0.98 | 17.44 | -1242.06 |
| $10^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.72 | -1155.42 |
| $11^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.56 | -1155.60 |
| $12^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.51 | -1155.66 |
| $13^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.49 | -1155.68 |
| $14^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $15^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $16^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $17^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $18^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $19^{th}$ | $\begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $20^{th}$ | $\begin{pmatrix} 0.04 \\ 00.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.48 | -1155.68 |
| $Convergence$ | $\begin{pmatrix} 0.04 \\ 0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.00 \\ 0.00 & 0.97 \end{pmatrix}$ | 0.93 | 26.56 | -1155.60 |
| $\star Actual$ | $\begin{pmatrix} 0.04 \\ -0.01 \end{pmatrix}$ | $\begin{pmatrix} 0.97 & 0.01 \\ 0.01 & 1.02 \end{pmatrix}$ | 0.93 | 27.61 | -1279.48 |
| $\star\star True$ | $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ | 0.95 | 30 | -1157.08 |

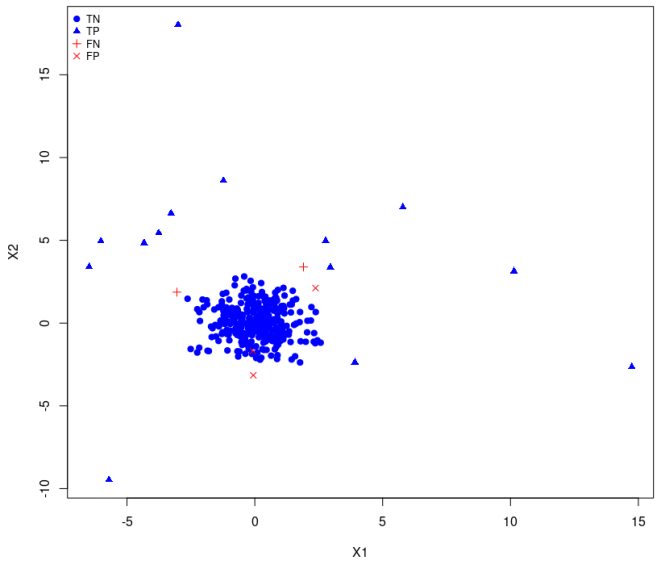
$\star$ denotes the values calculated with the observed data of the training set after running a M-step. $\star\star$ denotes the values with which data was simulated.

The ECM algorithm would halt at the $14^{th}$ iteration if the criterion were $l_i - l_{i-1} < 0.001$, and at the $11^{th}$ iteration using the Aitken criterion with a threshold of 0.001. Although the log-likelihood reaches a plateau at the $14^{th}$ iteration, terminating early at the $10^{th}$ would save computation time without compromising model performance in this particular case. In this study, the ECM algorithm stops after 10 iteration. The predictions of contamination information in the test set were obtained via running a E-step by plugging the parameter estimates obtained in the two CM-steps (using the training data) into Equations 2.23, 2.24 for $i = 1, \ldots, n$ and $g = 1, \ldots, G$ so that they can be compared to the truth.

Figure 2.1 displays the predicted contamination labels obtained at the $10^{th}$ step of the ECM for both the training and test sets. As anticipated, with $\eta = 30$, the majority of contaminated samples are noticeably distant from the cloud of non-contaminated samples. The ECM early stop algorithm demonstrates effectiveness in identifying contaminated samples in both sets. True positives, denoting correctly predicted contaminated observations, are represented by blue filled triangles, while true negatives are depicted by blue filled dots. Thus, observations filled with blue represent those with accurately predicted contamination labels, while those filled with red indicate misclassified observations. Notably, misclassified observations tend to be found near the non-contaminated samples. For instance, in the training set, the observation marked with the symbol '+' was erroneously classified as non-contaminated, despite being located a little distance from the edge of the non-contaminated cloud.




(a) Train set



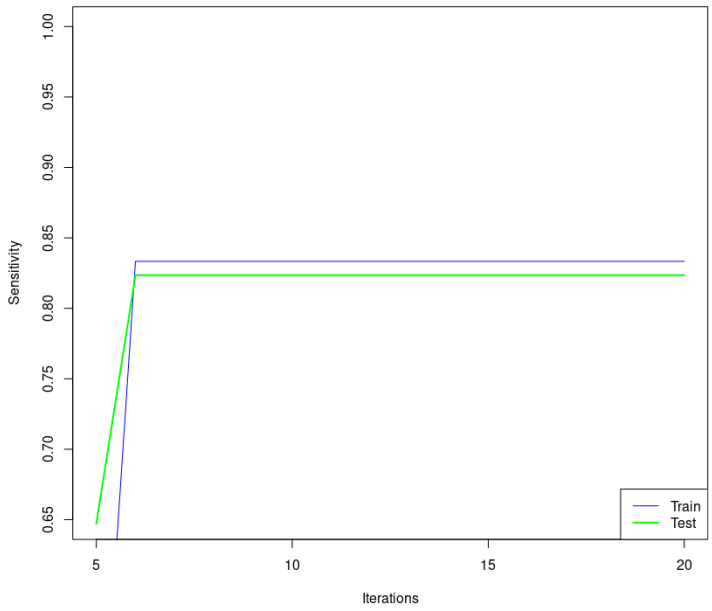

(b) Test set

Figure 2.1: Scatter plot, with predicted contamination labels calculated after stopping the ECM at the $10^{th}$ step for simulated data from a single class with 5% of contaminated samples.
TN correctly classified as non-contaminated, TP correctly classified as contaminated), FN wrongly classified as non-contaminated, FP wrongly classified as contaminated.

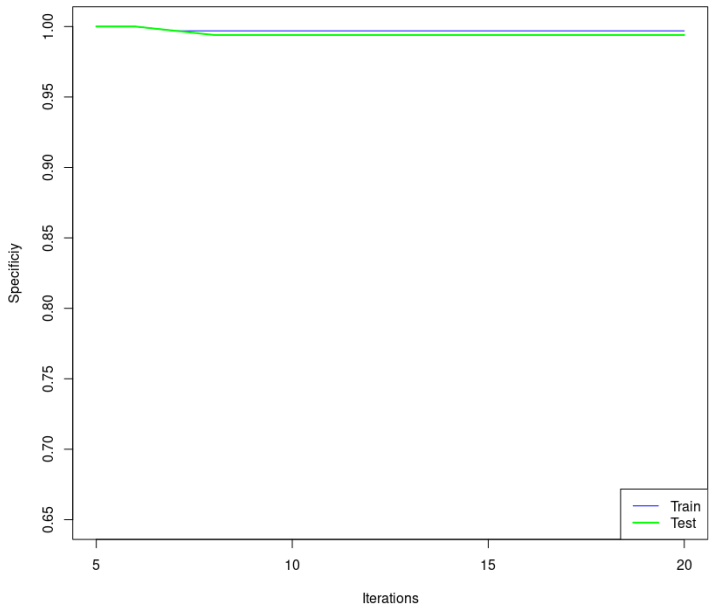
Contamination sensitivity (which is the ability of the model to correctly predict con-

taminated samples) is plotted in Figure 2.2. It seems to have stabilised at 0.95, which means identifying 95% of contaminated samples, after the $6^{th}$ iteration. In a similar way, contamination specificity (the ability to recognise non-contaminated samples) seems unchanged after the $6^{th}$ iteration of the ECM algorithm. These two metrics did not improve after the $6^{th}$ iteration.

Hence, the threshold for Aitken's convergence criterion was 0.001 and the ECM algorithm reached convergence at the $11^{th}$ iteration. These results suggest that it seems possible to stop the ECM algorithm earlier without deteriorating the predictions of contaminated labels. Overall, the model did a good job discriminating between contaminated and non-contaminated observations.




(a) Contamination sensitivity curve for the first 20 steps of the ECM algorithm in the train and test set




(b) Contamination specificity curve for the first 20 steps of the ECM algorithm in the train and test set

Figure 2.2: Contamination sensitivity and specificity for the first 20 steps of the ECM algorithm in the train and test set

Table 2.4: Confusion matrix showcasing predictions of contaminated labels obtained at the $10^{th}$ step of the ECM algorithm of a contaminated mixture of Gaussian distribution data with 5% of contaminated samples

| Actual | Predicted | | | |
|---|---|---|---|---|
| | Train | | Test | |
| | Contaminated | Non-Contaminated | Contaminated | Non-Contaminated |
| Contaminated | 20 | 4 | 14 | 3 |
| Non-Contaminated | 1 | 325 | 2 | 231 |

The parameter $\alpha$ controls the percentage of contaminated samples present in the data. In this scenario, $\alpha$ is small and consequently as Table 2.4 shows, the number of contaminated samples is 24 and 17 in the train and test sets respectively. In real data small levels of contaminated samples are expected. This is unbalanced data where the proportion of non-contaminated samples outnumber by far the contaminated samples. Because of this, accuracy is not the best option to measure the performance of the model identifying contamination. Performance metrics that adjust better to unbalanced data are sensitivity, precision, specificity, and F1 score. These performance metrics were described earlier in Section 2.3.2. Next, The issue of model selection is discussed as there is the need of choosing the "best" (in some sense) model from a range of possibilities (either different model parameteristations or variable sets or other differences).

## 2.8 Model selection

Hastie, Tibshirani and Friedman (2017) distinguish between model selection and model assessment, whereby model selection involves choosing an appropriate level of flexibility for a model, while model assessment focuses on evaluating the performance of a model (see Section 2.9.2).

Various criteria have been proposed to measure a model's suitability by balancing model fit and model complexity. Biernacki and Govaert (1999) compared the performance of different criteria to assess models for Gaussian model-based clustering and discriminant analysis and recommended the use of the Akaike information criterion (AIC) (Akaike, 1974) and Bayesian information criterion (BIC) (Adrian, 1996; Schwarz, 1978) (see also (James et al., 2013)).

The Akaike information criterion (AIC) and the Bayesian information criterion (BIC), among others statistics, are proposed as model selection criteria . The information criteria are typically defined for the large class of models fit by maximum likelihood. The AIC and BIC are given by the following expressions:

$$AIC = -2\ln(L) + 2k \tag{2.27}$$

$$BIC = -2\ln(L) + k\ln(n) \tag{2.28}$$

where $L$ represents the maximum value of the likelihood function of the assessed model, $n$ is the number of observations and $k$ the number of independent parameters in the model. The first term of the AIC tends to decrease as the number of parameters added to the model gets larger, however the second term $2k$ increases when more parameters are added to the model Burnham and Anderson (2004). Consequently, $k$ can be seen as a representation of the complexity of the model. The BIC is derived from a Bayesian perspective. Equation 2.28 shows that BIC statistics penalizes heavily models with many variables and will penalise them more than AIC when $\ln(n) > 2$. Small values in AIC and BIC suggests a model with low error, so the rule is to select models with the lowest AIC or BIC.

Model selection is usually used to choose between a set of candidate models on the same data. These models may differ in numerous way: variables used, complexity/parameterisations, transformations, etc. The first case arises usually when, in addition to fit a particular model to the data, there is the aim of looking at which variables to include in that model. This is known as variable selection. In the following section some of the common approach to variable selection are described. ‘

## 2.9 Variable selection

In the present era, companies have unprecedented access to extensive datasets obtained daily from smartphones, capturing owner localization, service experience ratings through electronic forms, and numerous other sources. This leads to datasets with high dimensionality both in terms of variables and observations recorded, providing ample opportunities for addressing various research questions. However, many of the collected features may

not contribute to addressing the key inquiries. In clustering and classification problems, it is often assumed that there exists a subset of variables capable of distinguishing between different classes within the data. Consequently, it is common to encounter numerous unnecessary variables that fail to provide meaningful information about the class structure. These are also known as noise variables.

Hastie et al. (2017) demonstrated through an example that the inclusion of non-informative variables can impair the performance of predictive models. Hence, there is a necessity to identify a concise subset of predictor variables that carry pertinent information. A similar situation in the context of clustering was discussed in Raftery and Dean (2006). When confronted with a large number of available variables, selecting the appropriate subset for the model becomes challenging.

Variable selection involves searching in a space of possible parameters. This search necessitates a set of variables to scrutinize, an initial starting point of the search (often a full or empty model), a direction (e.g. forward or backward), a stopping criterion, and a search strategy (e.g. greedy search or exhaustive search (Blum and Langley, 1997; Guyon and Elisseeff, 2003; Russell et al., 2022). Many of the developments in variable selection came initially in the context of linear regression models, where the aim is to model the relationship of a variable of interest based on a subset of possible explanatory variables. The variable selection problem can be seen as a special case of model selection problem, where each model in consideration is composed of a different subset of explanatory variables (George, 2000).

One of the additional advantages of having a model with a limited number of variables, apart from improving the modeling, is that it enhances interpretability and is also more cost-effective to maintain when collecting information on all variables is prohibitively expensive.

Various variable selection approaches have been proposed to mitigate the number of non-informative variables, which can be broadly classified into two categories: wrapper and filter methods (John G, 1994). These categories will be discussed in more detail in the following sections.

### 2.9.1 Filter methods

Filter methods assess variable importance independently of the prediction model by applying specific criteria. Variables meeting these criteria are then considered for inclusion in the model. For instance, in classification tasks, a filter method might employ a metric to gauge the relevance of the variable's association with the grouping variable. After individual assessment of variables, only those meeting the predefined condition are chosen to contribute to the classification model (Kuhn and Johnson, 2013).

Popular filter methods in the literature include correlation analysis, mutual information, chi-squared test, ANOVA, among others. For more details, refers to Hastie et al. (2017)

### 2.9.2 Wrapper methods

Wrapper methods are search algorithms that use variables as input to optimize a performance metric of the model. Unlike some methods, they do not assume a specific model structure beforehand, but instead rely on data and model performance metrics to find the optimal model. The algorithm iteratively adds and removes variables until further modifications no longer improve the performance metric. This iterative process allows for the evaluation of multiple models, ultimately identifying an optimal subset of variables that maximizes the model's performance.

The advantage of wrapper methods is that the selection criterion for an inclusion or exclusion of a variable is linked to the effectiveness or fit of the model. The disadvantage is that each addition or exclusion of a variable requires adjusting and often refitting the model and calculating the performance metric.

Raftery and Dean (2006) proposed a framework for partitioning the dataset at each stage of a wrapper algorithm. In these methods, the dataset $X$, comprising $p$ variables, is partitioned into three sets: $X^{(S)}$, $X^{(?)}$, and $X^{(O)}$.

- $X^{(S)}$: the set of already selected variables,
- $X^{(?)}$: the variable(s) to be proposed for inclusion into or exclusion from $X^{(S)}$ and,
- $X^{(O)}$: the set of variables composed for the remaining variables.

In the literature, methods such as forward selection, backward elimination, and stepwise algorithms are collectively referred to as stepwise methods. These methods, along with search algorithms like greedy search, are part of the wrapper models (Kuhn and Johnson, 2013). The subsequent sections will provide greater detail on stepwise methods, including the greedy search algorithm (Cormen T, 2022).

**Forward selection, backward elimination, and stepwise selection**

In forward selection, variables are incrementally added to the model, one at a time. Initially, the variable that offers the most noticeable improvement in the chosen performance metric is selected and included in the model. Subsequently, the remaining variables not yet included in the model are assessed for potential inclusion. At each step, the variable that results in improvement in the current model's performance metric is added to the model. Adding variables one at a time into the model continues until either all variables are added in the model and there are no more variables to add or when the condition to stop the search is satisfied (Derkseen and Kesselman, 265–282).

In backward elimination, the method starts with all p available variables being part of the model. In the next step one variable at a time is considered for elimination. The eliminated variable is the one that does not contribute to the chosen performance metric. Removal of variables stops when the model is composed of only one variable (or no variables in special cases) or because the stopping condition has been satisfied (Derkseen and Kesselman, 265–282).

In stepwise selection, the procedure combines forward selection and backward elimination (Derkseen and Kesselman, 265–282), potentially adding a variable at each iteration through a forward step and also a backward elimination step where variables that are currently part of the model are considered for dropping from the model.

**Greedy search algorithm**

An algorithm that has been proposed for finding the optimal subset of variables is a greedy search algorithm with a forward inclusion step . The greedy search algorithm adds one variable at a time in each iteration such that the variable added in each iteration is the one that guarantees the best performance on the selected performance metric $h(.)$ (see

Section 2.3.2 to see some of the common chosen metrics), where $\delta$ represents the desired performance or a threshold $h(.)$, which is the chosen performance metrics (e.g. correct classification rate, sensitivity, specificity, etc.), is expected to reach.

**Algorithm 1:** Greedy search algorithm

**Data:** $X, h, \delta$

**Result:** $X^{(S)}$

Initial partition: ;

$X^{(O)} \leftarrow \{X_1, X_2, \ldots, X_{p-1}, X_p\}$;

$X^{(?)} \leftarrow \{\}$;

$X^{(S)} \leftarrow \{\}$;

**while** $h(.) < \delta$ **do**

$j^* \leftarrow \underset{X^{(?)} \in X^{(O)}}{\arg\max}\, h(X^{(S)} \cup X^{(?)})$;

$X^{(S)} \leftarrow X^{(S)} \cup \{j^*\}$;

$X^{(O)} \leftarrow X^{(O)} \backslash \{j^*\}$;

**end**

The greedy search algorithm is one of the widely used variable selection methods due to its low level of complexity, and because it allows implementation of parallelization (Cormen T, 2022).

## 2.10 Summary

In summary, this chapter has reviewed some concepts of statistical learning such as different types of learning, labeled and unlabelled data, train and test sets, and model assessment metrics. It also gave an overview of finite mixture model in classification problems, the special case of a mixture of Gaussian models and the EM algorithm estimation for mixture parameters. In addition to this, the strategy to manage outliers in the statistics community was reviewed, with a focus the on including and modeling mild outliers, the available methods within the framework of finite mixture models for classification outlier problems, along with their limitations. An adaptation proposed by Punzo to address the limitation of not accommodating observations that differ from the reference model has been reviewed. The next part of this thesis focuses on addressing the limitation faced by contaminated mixtures of Gaussian models when dealing with high-dimensional data through the use of

a variable selection algorithm.

# Chapter 3

# Variable selection with supervised contaminated mixture of Gaussian models

## 3.1 Introduction

### 3.1.1 Previous work

In many real classification situations, the presence of contamination in a data set is very likely, and ignoring it will harm the parameter estimates and the prediction performance for class membership of new data. Typically, the information identifying contaminated samples is not present so it is not possible to exclude them or use them to help to identify the source of contamination which may also be a goal of the analysis. A versatile statistical model family that can be useful in classification problems is finite mixture models

A general limitation of mixture models is their susceptibility to being heavily influenced by a few atypical values (Campbell, 1984). Mixture model parameters are estimated using maximum likelihood estimation (MLE) via the EM algorithm, which is known to be very sensitivity to outliers in the data. To address this issue, Neykov et al. (2007) proposed a trimmed likelihood estimator (TLE) that removes observations considered unlikely to occur if the fitted model were true. Campbell (1984) suggested down-weighting the contribution of outliers during the parameter estimation stage. Punzo and McNicholas (2016) proposed addressing data containing mild outliers by modelling each group with a mixture

of two normal distributions, both centred at the same point in the sample, but allowing greater variability in the second component. Mazza and Punzo (2020) used mixtures of contaminated Gaussian distributions to enable mixtures of regression models to handle multivariate contaminated data. The advantages of using mixtures of contaminated Gaussians in regression models are that this method simultaneously identifies mild outliers in each cluster, estimates the clusters, and determines the corresponding cluster-specific regression functions given a set of independent variables and response variables from a sample of independent observations. Peel and McLachlan (2000) proposed a mixture of $t$-distributions to overcome the shorter tails of the normal distribution, as $t$-distributions provide longer tails that can accommodate observations with zero probability in a normal distribution. Consequently, mixture models of contaminated Gaussian distributions inherit this limitation, which is something to bear in mind.

While the implementation of contaminated Gaussian mixture models is suitable for modeling contaminated data, an additional challenge arises. Specifically, when the (variable) dimensionality of the data is extremely high (particularly when it exceeds the number of observations), the estimates of the variance-covariance matrix may become singular or nearly singular (Naderi, 2024). This means that less complex and poorer fitting models may be the only option in modelling the data. In addition not all variables in such high dimensional cases are useful for classifying observations or identifying contamination. Including these non-informative variables not only causes problems for fitting models, it can also degrade classification performance. Therefore, a procedure that identifies the non-informative variables to exclude them and select the informative variables to build a model is convenient.

Although variable selection has been extensively investigated for supervised learning and also for model-based clustering, less work has been done in the scenario of contaminated data. Murphy et al. (2010) developed a wrapper for variable selection whith a greedy and headlong search algorithm to find a local optimum in the model space for semi-supervised discriminant analysis. This work was based on a previous approach introduced by Dean (2006) that considered potential correlations between separating and non-separating variables for first time. In a similar way Maugis (2011) proposed a variable selection procedure that partitioned variables into relevant and irrelevant through a two

forward step algorithm. Celeux et al. (2019) introduce regularization to overcome the slowness of the stepwise algorithm when dealing with data in high dimensions. Other wrapper methods have been introduced, but they are not designed to deal with outliers in the data. The wrapper methods mentioned are designed for mixtures of Gaussian distribution assuming the absence of outliers. Consequently, the contaminated mixture of Gaussian models has received less attention in comparison with the mixture of Gaussian models. A mixture of a contaminated Gaussian model with an unrestricted variance-covariance matrix is a highly parametrized model. One approach to extend the mixture of contaminated Gaussian to high dimensional data was introduced by Punzo (2018) and relies on a mixture of contaminated factor analyzers that creates $q$ latent variables where $q < p$. This approach creates parsimonious models reducing the dimension of the original data set, but it sacrifices interpretability by creating latent variables.

Cappozzo et al. (2021) introduce two variable selection approaches that have some similarities with the proposed model, but these approaches have different aims. One of the variable selection methods uses a greedy search algorithm with a forward and backward steps to assess whether the proposed variable provides group information by comparing two models; one model including the variable and another without the variable. The comparison of these two models is done using the trimmed Bayes information criterion. The main difference of these procedures with our proposed approach is that the former intends to remove outliers instead of accommodating them into the model. The other approach uses the maximum likelihood subset selector theory developed for clustering and also discards observations that are considered as outliers. The approach presented in this work intends to accommodate mild outliers in the model rather than discard them.

### 3.1.2 New work

In this chapter, the challenge of extending the mixtures of contaminated Gaussian distributions to model high dimensional contaminated data is addressed. The main idea of the proposed approach tailored a variable selection algorithm for the supervised (classification) mixtures of contaminated Gaussians model. The primary aim is to reduce the dimensionality of the data, presuming that a small subset of the available variables holds crucial information regarding the class membership and contamination status of the sam-

ples. The objective is to pinpoint this subset of variables and integrate them into the supervised contaminated mixture model. However, in cases where the number of variables $p$ is substantial, the potential number of subsets to be assessed balloons to $2^p$, as each variable can either be included or excluded from the total subset under scrutiny. For large $p$, this can become unmanageable, as the array of possible variable subsets to be considered can be huge.

Several aspects of variable selection need consideration to effectively identify the most informative variables for predicting class information and contamination status. These aspects include determining the search strategy for the variable selection, the direction of search, establishing a stopping criterion for the search and selecting an appropriate metric to evaluate algorithm performance. Additionally, besides evaluating variable selection based on prediction performance, it is important to assess the frequency of non-informative variables included in the selected subset and the occurrence of true variables being selected by the variable search algorithm. This can be done using simulation studies where the truth is known.

The methodology of the variable selection in the supervised context of contaminated normal mixtures is introduced in Section 3.2.2. An adaptation of a greedy search algorithm customized specifically for supervised mixtures of contaminated Gaussian models is explained in Section 3.2.1. This proposed approach is evaluated on its selection of a subset of variables that can separate the classes and identify contaminated observations at the same time through simulations. The simulation framework in which the proposed method will be evaluated and the method of measuring its performance is described in detail in Section 3.2.2. Additionally results of applying the proposed approach to simulated scenarios and a model elucidating the effects of certain factors defined in the simulation framework on performance metrics is presented. An analysis of the frequency of inclusion of informative and non-informative variables is provided in Section 3.3. The outcomes of implementing the proposed method to fit plasmode datasets Gadbury et al. (2008), formed by introducing simulated contaminated samples to actual datasets, are discussed in Section 3.4. In Section 3.8, the advantages and limitations of the proposed method are examined, along with a discussion of other research efforts aimed at extending the multivariate mixture of contaminated Gaussian distribution to higher dimensions.

## 3.2 Methodology

In this section we discuss the addition of variable selection to the mixture of contaminated Gaussians model in order to remove non-informative variables. First we discuss a forward greedy search algorithm to wrap around the mixture model for exploring the variable space.

### 3.2.1 Tailoring a forward greedy search algorithm for a supervised mixtures of contaminated Gaussian models

The general greedy search algorithm was introduced in Section 2.9.2. One of the initial queries raised pertains to the starting point for the search. Given the challenge of dealing with numerous observed variables, initiating a forward search seems reasonable, starting with an empty model and slowly include more variables.

When considering which search strategy to employ, there might be a temptation to test all possible variable combinations to determine the optimal set for class differentiation and identification of contaminated observations. Conducting such an exhaustive search with a lengthy list of variables would consume excessive time and prove inefficient. Nevertheless, a less ambitious search strategy, known as a hill-climbing strategy or greedy search, relies on performance metrics to add one variable at a time in a stepwise manner. In contrast to the exhaustive search strategy, greedy search does not find the best solution but rather a local optimum solution (Kohavi and John, 1997).

The variable selection problem, where all variables are initially considered as potential class-separation variables, transforms into a model selection problem. The search for variables proceeds in steps, denoted by $t$, where each step $t$ involves partitioning the set of variables in the training data into three sets of variables $X^{(S)}, X^{(?)}$, and $X^{(O)}$ (as defined in Section 2.9). Here, $D^{(t)}$ represents the number of variables in the set $X^{(O)}$. At each step $t$, there are $D^{(t)}$ possible models competing, and the chosen model is the one with the highest improvement in the chosen performance metric on the test set. Consequently, at any stage, if there exists a variable that produces the highest improvement in the chosen performance metric on the test set, it is included in the set $X^{(S)}$ and excluded from the set $X^{(O)}$. If no variable offers improvement the search algorithm is stopped. In this work, the chosen performance metric is true correct classification rate, which is calculated on

the test set, due to its simple calculation and interpretation.

Here, Figure 3.1 shows the adaptation of the forward greedy search algorithm for a supervised model that uses contaminated mixtures of Gaussian distributions for the case of continuous data. This adaptation assumes that the number of classes and the label information of observations in the training set are known.

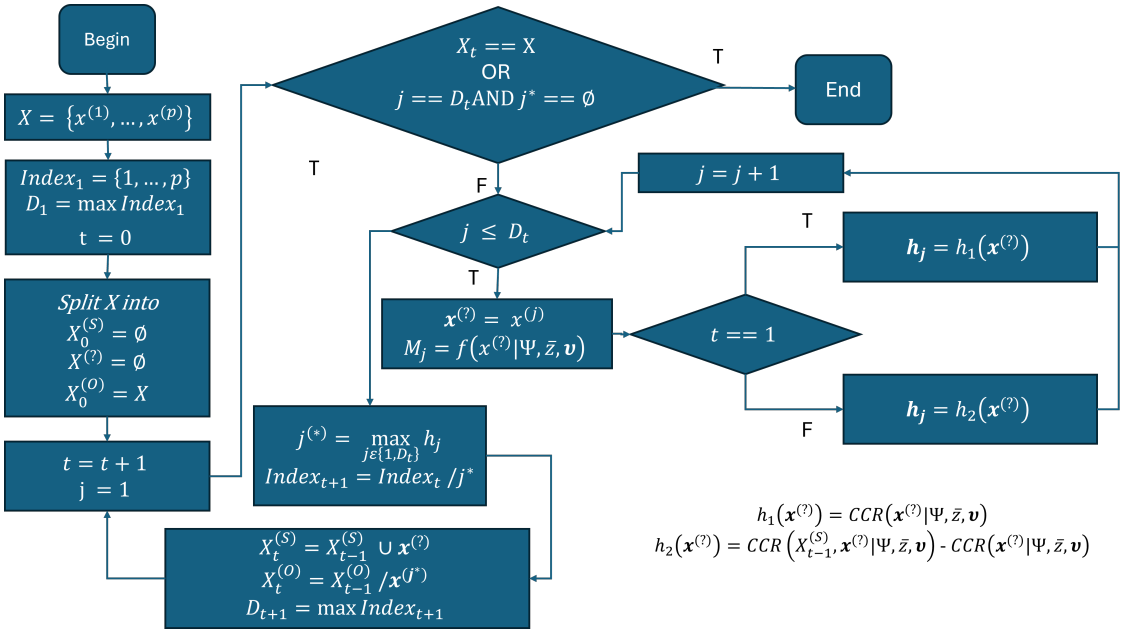



Figure 3.1: Tailored forward greedy search for supervised contaminated mixtures of Gaussian distributions

- **Initialization:** Initially the set of variables in the training data is partitioned in the following way: $X^{(S)} = \emptyset, X^{(?)} = \emptyset$, and $X^{(O)} = X$. This means that at the beginning the subset of the "selected variables" and the proposed variable are empty. Moreover, the subset of other variables starts containing all the available variables in the training set.

- **First step:**

  To initialize the model, a greedy search is carried out to find the best model of size 1. At this step, $D^{(1)} = p$, since $X^{(O)} = (X^{(1)}, \ldots, X^{(D_1)})$ and the total number of models of size 1 is $p$. Subsequently, the covariance structures from Table 2.2 that best suit their corresponding models $M_1, \ldots, M_{D^{(1)}}$ in the training set are chosen. The model with the highest class *Correct Classification Rate (CCR)* in the test set

is then chosen.

$$\begin{aligned} M_1 &= f(X^{(1)}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \\ M_2 &= f(X^{(2)}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \\ &\vdots \\ M_{D^{(1)}} &= f(X^{(D^{(1)})}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \end{aligned} \tag{3.1}$$

where $f(X^{(j)}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu})$ is the contaminated normal mixture model with the best covariance structure chosen for that variable $X^{(j)}$

The selected variable recorded in $X_t^{(S)}$ at the first step, $t = 1$, is the variable that gives the highest value of class CCR calculated on the test set:

$$X_1^{(S)} = \underset{X^{(?)} \in X^{(O)}}{\arg\max}\, CCR(f(X^{(?)}|\boldsymbol{\psi}, l, \boldsymbol{\nu}))$$

The subset $X_1^{(O)}$ is then updated, excluding the variable chosen to be part of the selected variable subset $X_1^{(S)}$, and the CCR of the model is recorded as the baseline for considering the next expansion of the model.

$$X_1^{(O)} = X \backslash \{X_1^{(S)}\}$$

- **General inclusion step:** At step $t$, the proposed variable, $X^{(?)}$, checked for inclusion into $X_t^{(S)}$ is each single variable currently in the set of other variables $X_{t-1}^{(O)}$. The new variable to be included in $X_t^{(S)}$ is selected based on the greatest improvement brought to the model formed by the currently selected variables $X_{t-1}^{(S)}$ in class separation within the test set (maximized over the possible covariance parameterizations considered using the training set). At step $t$, there are $D^{(t)}$ potential models:

$$\begin{aligned} M_1 &= f(X_{t-1}^{(S)}, X^{(1)}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \\ M_2 &= f(X_{t-1}^{(S)}, X^{(2)}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \\ &\vdots \\ M_{D^{(t)}} &= f(X_{t-1}^{(S)}, X^{(D^{(t)})}|\boldsymbol{\psi}, \tilde{\mathbf{z}}, \boldsymbol{\nu}) \end{aligned} \tag{3.2}$$

The model with the highest gain in CCR is selected. The corresponding variable is then our proposed variable for inclusion into the set of selected variables. As long as this gain is positive, the partition of the set $X$ is updated to include this proposed variable in the set of selected variables, and the CCR of the selected model is recorded.

$$
\begin{aligned}
&CCR_{diff}(X^{(?)}) = (CCR(f(X_{t-1}^{(S)}, X^{(?)}|\boldsymbol{\psi}, l, \boldsymbol{\nu})) - CCR(f(X_{t-1}^{(S)}|\boldsymbol{\psi}, l, \boldsymbol{\nu})))\\
&\text{if } \max(CCR_{diff}(X^{(?)})) > 0 \text{ then}\\
&X_t^{(S)} = \{X_{t-1}^{(S)}, \underset{X^{(?)} \in X^{(O)}}{\arg\max}\, (CCR_{diff}(X^{(?)}))\\
&X_t^{(O)} = X_{t-1}^{(O)} \backslash \{X^{(?)}\}
\end{aligned}
\tag{3.3}
$$

- The general inclusion step is iterated after the first step until either all variables in the subset $X^{(O)} = X^{(1)}, \ldots, X^{(D_{t+1})}$ have been proposed and rejected because they do not improve the CCR over the currently selected set of variables or all variables have been included in the subset $X^{(S)}$.

### 3.2.2 Simulation framework

Let us revisit the scenario described in Section 2.2, where there are $G$ classes within the data, and the class membership is only available for $m$ observations. The matrix $X$ contains $n$ $p-$dimensional vectors, which are the observations, and the variables that differentiate the classes are known. The task is to construct a model capable of distinguishing observations between classes and additionally identifying whether the observation is contaminated. There is interest in studying the built model's behaviour in different datasets and to explore under which conditions it performs better.

Several factors can impact the performance of a classification model, including the number of classes, class proportions (Japkowicz and Stephen, 2002; Weiss and Provost, 2001), variances, number of observations, number of variables, observations in the training set, pairwise correlation (Derkseen and Kesselman, 265–282), distance between classes (Steinley, 2003), and number of variables that distinguish between classes. To evaluate the influence of some of these factors on model performance, a simulation study framework was constructed.

Three approaches are being assessed on the simulations, each comprising modeling on different sets of variables: "True variables", which are the known variables that create a separation between classes, "Selected variables", which are the variables identified via a search algorithm; and "All variables", which are all observed variables. The objective

was to compare the performance of a variable selection procedure with the other two approaches. The study, consisting of 10 simulated replicates for each of the possible scenarios, controlled differing levels of the aforementioned factors.

In each simulated replicate, the *correct classification rate (CCR)*, sensitivity and specificity were recorded for **class membership and contamination labels**. Details of the proposed simulation framework for this study, including the various values that the factors can take, are provided in Table 3.1.

Table 3.1: Set of different factor values in simulation framework

| Factors | Description | Levels |
|---|---|---|
| V | Set of variables to train the model | True, all, selected |
| $F_1$ | The distance between mean classes | Very overlapping (VO) .<br>Medium distance (MD).<br>Very distant (VD). |
| $F_2$ | Number of classes | 2, 3 |
| $F_3$ | Class proportion | Balanced (50%-50% or 33%-33%-33%).<br>Imbalanced (90%-10% or 60%-20%-20%). |
| $F_4$ | Number of variables | 5, 100 |
| $F_5$ | Percentage of samples used as training | 75%, 85% |
| $F_6$ | Correlation structure | Strong correlation between separating variables (SCBSV)<br>Strong correlation between separating and not separating variables (SCBSNSV)<br>Strong correlation between not separating variables (SCBNSV)<br>Independence (IND) |
| $F_7$ | Percentage of non-contaminated samples | 80%, 90% for 2 classes<br>or 80%, 90% and 90% for 3 classes |
| $F_8$ | Variance inflation factor | 5, 30 for 2 classes<br>5, 30 and 30 for 3 classes |
| $F_9$ | Number of separating variables | 2, 3 |

All simulations had 3000 observations

† For factor F1: (VO) denotes a separation in Euclidean distance of $2.1\sigma$ between class means, (MD) a separation in Euclidean distance of $4.2\sigma$, and (VD) denotes a separation in Euclidean distance of $8.5\sigma$.

For Factor F6: strong correlation means $\sigma_{i,j} = 0.8$, and independence denotes $\sigma_{i,j} = \sigma_{j,i} = 0$.

The factor $V$ encompasses the different set of variables ($V$) that are going to be compared, "true variables", "selected variables", and "all variables". This is obviously how

we assess performance of the variable selection. All simulations were generated with 3000 observations to give sufficient information to allow all covariance structure mixture models to be fitted.

The first factor, often the most influential among the selected variables (Steinley, 2003), is the distance between mean classes. Let's consider a variable that follows a standard Gaussian distribution with a mean of 0 and a standard deviation of 1. It is widely accepted that approximately 68.27% of the observations fall within 1 standard deviation of the mean, while around 95.45% fall within two standard deviations, and nearly 99.73% fall within three standard deviations. With this in mind, it is often practical to represent two classes derived from two multivariate Gaussian distributions, each with different means but sharing the same variance-covariance matrix—an identity matrix of dimension 5.

For simplicity, the mean of the first class is positioned at the origin $(0, 0, 0, 0, 0)$. The other class can be constructed with two separating variables $(e.g. X_2, X_4)$ positioned at a distance of $c$ standard deviations from the origin, while the remaining three variables are set at the origin. Consequently, the mean of the second class is positioned at $(0, c, 0, c, 0)$. Thus, the closer $c$ is to the origin, the closer the two classes will be, resulting in a greater level of overlap between them. Therefore, one potential value for $c$ in scenarios where two classes mapped in 5 dimensions require a significant degree of closeness between mean classes ("very overlapping (VO) classes"), but without reaching the extreme where the two groups are impossible to separate, is 1.5 standard deviations from the origin for the separating variables. This leads to the construction of two classes with a Euclidean distance between their mean of $\sqrt{(0-0)^2 + (0-1.5)^2 + (0-0)^2 + (0-1.5)^2 + (0-0)^2} \approx 2.1$ standard deviations.

Similarly, in scenarios where two classes need to be built with median distance (MD) and very distant (VD) from each other, a possible value for $c$ would be 3 and 6 respectively. For scenarios with three separating variables $(e.g. X_2, X_4, X_5)$, one option is to place the mean of the second class at coordinates $(0, c, 0, c, c)$ and choose $c$ accordingly. For cases with three groups and two separating variables $(e.g. X_2, X_4)$, the mean of the second class could be placed at coordinates $(0, c, 0, c, 0)$, and the mean of the third class at coordinates $(0, -c, 0, -c, 0)$. In scenarios with three groups and three separating variables $(e.g. X_2, X_4, X_5)$, the means of the second and third classes can be placed at coordinates $(0, c, 0, c, c)$ and $(0, -c, 0, -c, -c)$ respectively, with $c$ chosen as before to achieve the desired degree of distance between mean classes. The distance between mean classes is tested

at three levels: "very overlapping classes (VO) with $2.1\sigma$ distance between mean classes, medium distance (MD) with $4.2\sigma$ distance, and very distant (VD) with an $8.5\sigma$ distance. The second factor, the number of classes (James et al., 2013; Tibshirani et al., 2001), was assessed at two levels: 2 and 3. The third factor, class proportion, was examined under two conditions: balanced, where classes have roughly equal number of observations (split evenly for 2 and 3 classes), and imbalanced, with one class having the majority of observations (90% for the first class and 10% in the case of two groups, and 60%, 20%, and 20% for three groups).

The fourth factor, number of variables, was tested at two levels, 5 and 100. The fifth factor, percentage of observations assigned to the training set, was evaluated at two levels: 75% and 85%. The sixth factor, correlation structure, was assessed at four levels: strong correlation between separating variables (SCBSV), strong correlation between separating and non-separating variables (SCBSNSV) , strong correlation between non-separating variables (SCBNSV), where strong correlation was given as $\rho_{i,j} = 0.8$, and independent variables (IND) where $\rho_{i,j} = 0$.

The seventh factor was the percentage of non-contaminated samples, evaluated at two levels 80% for the first class and 90% for the second class in the case of two groups, and $80\%, 90\%$, and 90% in the case of three classes. The eighth factor, variance inflation factor, was set at 5 and 30 for two classes and $5, 30$, and 30 for three classes. The ninth factor, the number of separating variables, was evaluated at two levels: 2 and 3. Each setting involved 10 simulations, resulting in a total of 384 simulated settings and $3,840$ simulated datasets. As in the $3,840$ simulations eighth and ninth factors were fixed for two and three classes at the values detailed before, and additional number of simulations were run to assess these factors at an additional level at each class. These additional simulations were run for two and three classes mapped only in 5 dimensions due to computation time. In this additional simulations, for two classes the eighth factor, the percentage of non-contaminated samples was evaluated at 90% for the first class and 80% for the second class in the case of two groups, and 90%, 80%, and 80% in the case of three classes. The eighth factor, variance inflation factor was assessed at 30 and 5 for two classes and 30, 5, 5 for three classes.

An illustration of recording one performance metric for all replicates of a particular scenario is shown in Table 3.2. At each replicate of the simulated scenario true, selected, and all variables approaches are to be compared and the response variable is to be one of

the performance metrics.

Table 3.2: Test class correct classification rate (CCR) for 10 replicates of two very overlapping and balanced classes with correlated non separating variables mapped in 100 dimensions

| | Scenario | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Set of Variables | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| True variables | 0.86 | 0.82 | 0.82 | 0.85 | 0.81 | 0.84 | 0.85 | 0.84 | 0.84 | 0.84 |
| Selected variables | 0.90 | 0.86 | 0.88 | 0.89 | 0.87 | 0.88 | 0.89 | 0.87 | 0.90 | 0.89 |
| All variables | 0.90 | 0.85 | 0.88 | 0.90 | 0.86 | 0.88 | 0.89 | 0.87 | 0.89 | 0.89 |

Before evaluating the performance on simulation factors generally, in the next section a visualisation of one scenario is presented to give intuition as to what the simulated datasets look like as well as how the results are achieved.

### 3.2.3 Two very distant and balanced classes with strong correlated non separating variables mapped in 5 dimensions with 2 separating variables

In this simulation, there are 3000 simulated observations mapped in 5 dimensions. The data were simulated from two contaminated Gaussian distributions, each representing one class. The mean for the first class was located at the origin $(0, 0, 0, 0, 0)$, while for the second class was $(0, 6, 0, 6, 0)$. Thus, there were two separating variables, $X_2$ and $X_4$ and the remaining variables were non-informative as they did not contain any class information. The proportions of non-contaminated samples for the first and second class were 80% and 90% respectively, and their inflation factors were 5 and 30 respectively. The number of observations in the training set was 2250 (75%), and the remaining 750 were in the test set. Both classes have the same covariance matrix with a strong correlation between two non-separating variables, $\sigma_{1,3} = \sigma_{3,1} = 0.8$, and each of the five variables with variance $\{\sigma_j\}_{j=1}^{5} = 1$.

In Figure 3.2, the first and second classes in the test set are represented by the colours blue and green, respectively. Additionally, in the second column and fourth row of the pairs

plot a linear pattern is visible, revealing the strong correlation between non-separating variables $X_1$ and $X_3$. The two classes are well separated on the pair of variables $X_2$ and $X_4$, hence it is expected that the greedy search algorithm identifies these two variables that distinguish between classes.

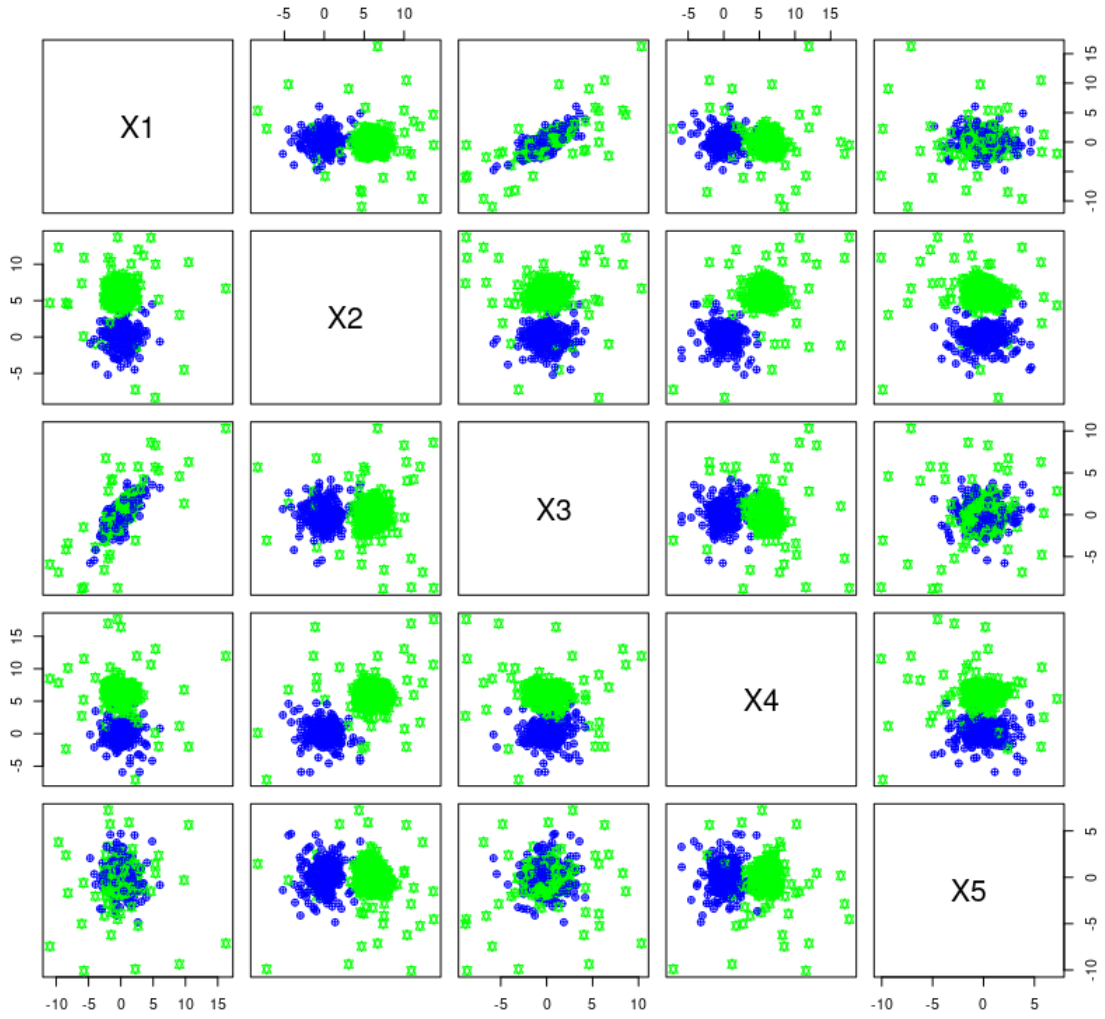



Figure 3.2: Coloured pairs plot of two very distant balanced classes with correlated non-separating variables ($\sigma_{1,3} = \sigma_{3,1} = 0.8$) mapped in 5 dimensions with 2 separating variables in the test set with colour denoting class

The greedy search algorithm selected the variables $X_2, X_4$ and $X_5$ to construct the model, achieving a test class correct classification rate of 99%. Table 3.3 illustrates the step-by-step progress of the greedy search. Initially, there were 5 potential variables for selection. However, $X_4$ was chosen in the first step as it offered the greatest level of test

class correct classification rate (CCR) and was consequently included in the model. In the subsequent step, $X2$ was proposed as the best variable from the remaining variables not yet included in the model. As it increased the test class CCR, it was accepted into the set of selected variables. Finally, in the third step, step the best variable $X_5$ improved the test class CCR and so was accepted for inclusion into the selected variables and at this point, the algorithm stopped.

Table 3.3: Progress of the greedy search algorithm for a simulation of two very distant balanced classes with correlated non-separated variables mapped in 5 dimensions in the test set

| Step No | Proposed Variable | Covariance Structure | CCR Class (Test set) | Result |
|---|---|---|---|---|
| 1 | $X_1$ | E | 0.52 | Not Included |
| | $X_2$ | E | 0.96 | Not Included |
| | $X_3$ | E | 0.52 | Not Included |
| | **$\mathbf{X}_4$** | **E** | **0.97** | **Included** |
| | $X_5$ | E | 0.51 | Not Included |
| 2 | $X_1$ | EII | 0.97 | Not Included |
| | **$\mathbf{X}_2$** | **EII** | **0.98** | **Included** |
| | $X_3$ | EII | 0.98 | Not Included |
| | $X_5$ | EII | 0.97 | Not Included |
| 3 | $X_1$ | EII | 0.98 | Not Included |
| | $X_3$ | EII | 0.98 | Not Included |
| | **$\mathbf{X}_5$** | **EII** | **0.99** | **Included** |

If it worth considering what would happen if the model were constructed solely with the true variables or with all the variables. To explore this question, a comparison of the three different sets of variables to build a model is considered. According to Table 3.4, there appears to be no distinction among models in accurately classifying new observations into their respective classes, as the test class correct classification rate, test class sensitivity, and test class specificity exhibit similarity across the models.

Table 3.4: Performance metrics for predicting class labels comparing variable selection with two other approaches for two very distant balanced classes with correlated non separated variables mapped in 5 dimensions in the test set.

| Set of Variables | Test class CCR | Test class sensitivity | Test class specificity |
|---|---|---|---|
| True variables | 0.98 | 0.98 | 0.98 |
| Selected variables | 0.98 | 0.98 | 0.98 |
| All variables | 0.99 | 0.99 | 0.99 |

Note: Sensitivity and specificity are calculated for each class and then averaged.

The performance metrics, which measure the models' ability to discriminate contaminated from non-contaminated observations, are given in Table 3.5. While the compared models exhibit no big differences in test contamination correct classification rate (CCR) and test contamination specificity, there is a discrepancy in test contamination sensitivity. The model incorporating all variables outperforms the one formed by only the true variables and the selected variables. Specifically, the model comprising solely the true variables identifies 35% of the contaminated samples, while the model constructed with selected variables detects 55%, and the model using all the variables can identify only 76% of them. It is worth noting that these results are based on one replicate of a scenario, so it is reasonable to consider potential variations in the other scenarios where different factors are varied (see Table 3.2.2).

Table 3.5: Performance metrics for predicting contamination labels comparing variable selection with two other approaches for two very distant balanced classes with correlated non-separated variables mapped in 5 dimensions in the test set.

| Set of Variables | Test contamination CCR | Test contamination sensitivity | Test contamination specificity |
|---|---|---|---|
| True variables | 0.87 | 0.37 | 1.00 |
| Selected variables | 0.89 | 0.55 | 0.98 |
| All variables | 0.91 | 0.76 | 1.00 |

There are four characters (•, ▲, ×, and +) to represent the four possible outcomes. Observations that are correctly classified in the first class are associated with the "•" symbol,

the observations correctly classified in the second class are associated with the "▲" symbol, the observations incorrectly predicted in the first class are associated with the "×" symbol, and the observations incorrectly predicted in the second class are associated with the "+" symbol. The model formed by the selected variables $X_2, X_4$, and $X_5$ discriminates quite accurately the class of new observations. The first class is coloured in light blue and the second class in light green and it is noticeable that there are a few misclassified observations that are coloured in orange as Figure 3.3 shows. The vast majority of these few misclassified observations (coloured in orange) belongs to the first class, but they are located at the edge of the cloud of points of the first class.

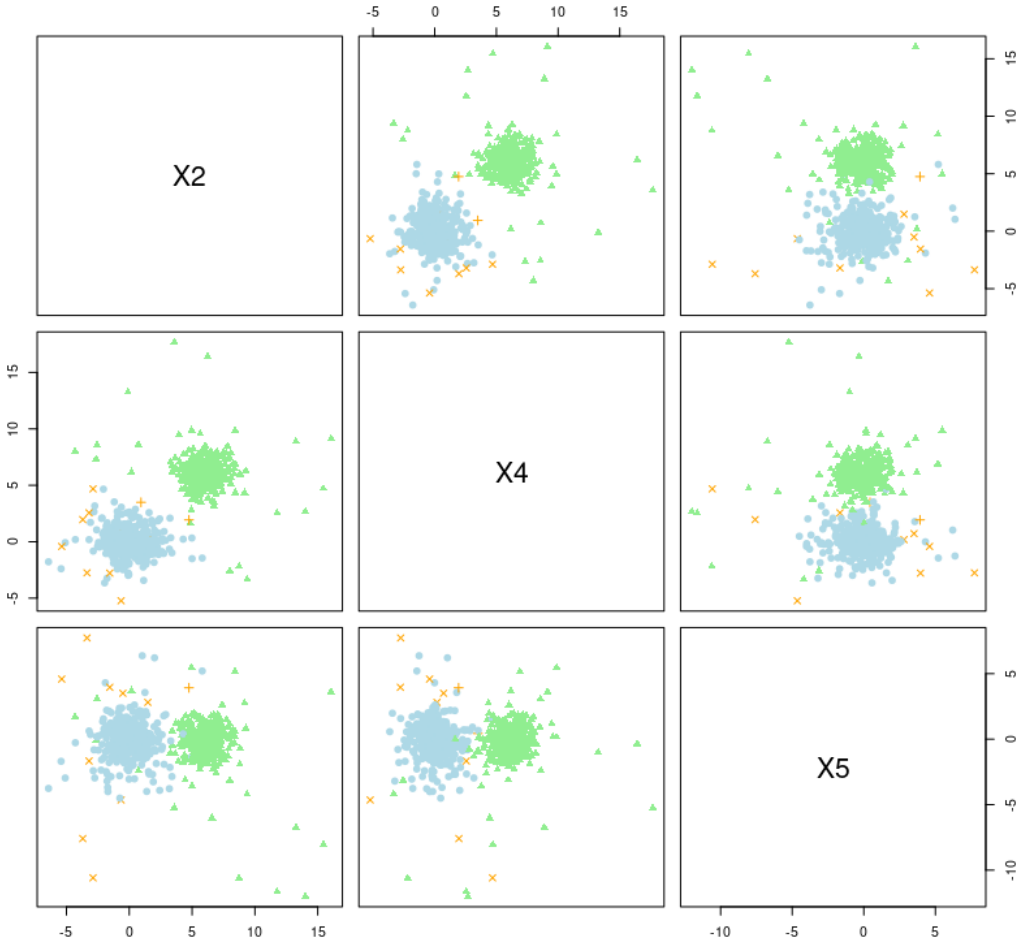



Figure 3.3: Misclassified class labels for observations for two balanced classes with two separating variables and three no separating variables strongly correlated on the test set

Although the model composed of the variables obtained by the greedy search algorithm

discriminates quite well between these two classes, it has difficulties in discriminating whether an observation is contaminated. There are six characters ($\bullet, \blacktriangle, \square, \times, +, \circ$, and $\star$) to represent six possible outcomes. Observations from either the first and second class that are correctly predicted as non-contaminated (true negatives TN) are associated with the "$\bullet$" symbols; the observations belonging either the first or second class correctly predicted as contaminated (true positives TP) are associated with the "$\blacktriangle$" symbols; the observations belonging to the first class and incorrectly predicted as non-contaminated (false negatives FN) are associated with the "$\times$" symbol; the observations from the first class incorrectly predicted as contaminated (false positives FP) are associated with the "$+$" symbol; the observations from the second class wrongly predicted as non-contaminated (false negatives FN) are associated with the "$\star$" symbol; and observations from the second class wrongly predicted as contaminated (false positives FP) are associated with the "$\circ$" symbol. The first class is coloured in light blue, the second class in light green, observations from the first class whose contamination labels were wrongly predicted are coloured in brown, and those observations from the second class whose contamination labels were incorrectly predicted are coloured in purple.

In Figure 3.4 there is a plot of false positives and false negatives for the prediction of contaminated observations showing that in contrast to class prediction where there was a clear class separation with the variables selected by greedy search, the selected variables are not enough to discriminate between contaminated and non-contaminated observations. The model formed by the selected variables $X_2, X_4$, and $X_5$ discriminates quite accurately the class of new observations.

However, the model struggles in detecting contaminated samples in the first class since the vast majority of observations with wrong prediction of their contamination labels are from the second class. It could be due to the substantial inflation factor that regulates the dispersion of the observations in the second class, resulting in some of the second-class observations being situated within the cloud of non-contaminated observations from the first class. Furthermore, the inclusion of an extra variable could enhance the performance of the model comprised of $X_2, X_4$, and $X_5$ in identifying contaminated observations.

The pair $X_1$ and $X_3$ implies that while individual variables such as $X_1$ or $X_3$ may not

help to improve the model's performance in class prediction, either of them could slightly contribute to improving the model's ability to identify contaminated samples.

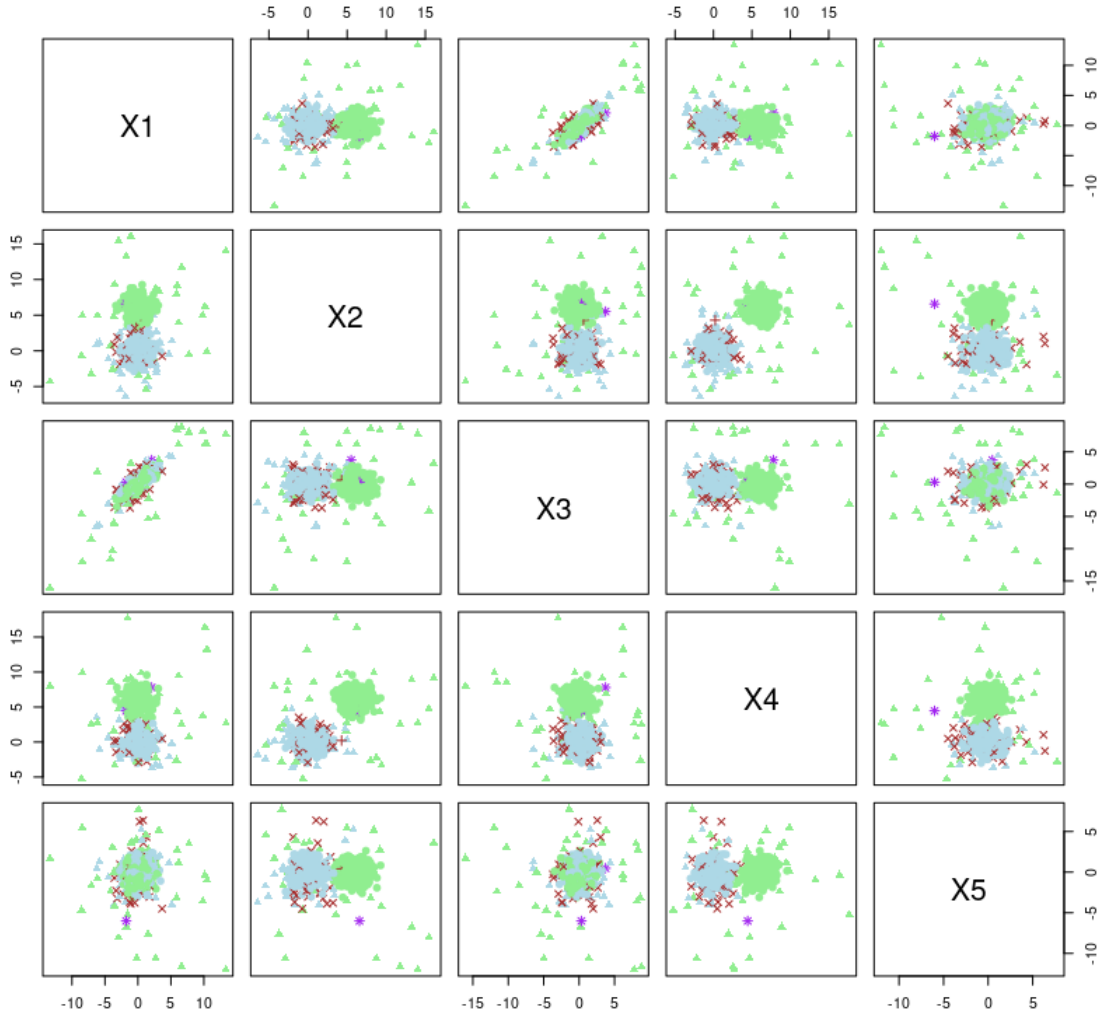



Figure 3.4: Misclassified contamination labels for two balanced classes with two separating variables and three no separating variables strongly correlated on the test set

## 3.3 Simulation studies

The evaluation of the variable selection approach includes comparisons of variables selected and their performance with two alternative subsets: utilizing all variables (assuming a full model fit) and possessing knowledge of the true informative variables that separate classes (a hypothetical scenario). These comparisons are conducted across datasets featuring very overlapping, medium, and very distance between mean classes, varying balanced

and unbalanced classes, number of clusters, and other relevant factors as specified in the previous Section 3.2.2.

### 3.3.1 Scenarios with fixed distance between class means factor and other factors varied

The preceding section showcased the outcomes of a single replicate within a scenario featuring two markedly distant classes, mapped in a lower dimension (5 dimensions). This prompts inquiries into the performance of the proposed method under varying circumstances, such as distance between class means, classes mapped in higher dimensions, an augmented number of non-informative variables, or other alterations.

It is widely acknowledged that class overlap poses a significant challenge in classification tasks (Duda et al., 2000; James et al., 2013). Therefore, the assumption is made that if the proposed approach fares well in a challenging scenario, it would likely excel in less arduous ones. In this scenario the factor $F_1$ is held constant while the number of classes $F_2$, class proportion $F_3$, number of variables $F_4$, percentage of samples used in training $F_5$, and correlation structure $F_6$, are varied. In all simulations 3000 observations were generated for each setting obtained by varying the factors mentioned at each of their levels and 10 replicate datasets were created for each combination.

In Figure 3.5, differences in test class correct classification, test class sensitivity, and test class specificity for comparison between models shown in the first row in the plot reveal that the model with a variable selection procedure *SM* performs better than the true model *TM* and is very similar to the model including *all* the available variables. However, looking at the second row of the boxplots, it is clear that although the "selected variables" subset outperforms the "true variables" subset regardless of the distance between class means, its test contamination sensitivity is lower than a model composed of all the variables. Additionally, there is not a big difference between models in terms of specificity, and the higher variability in the interquartile range in sensitivity is due to scenarios where the "true variables" subset and occasionally the model with variable selection generated models with poor sensitivity.

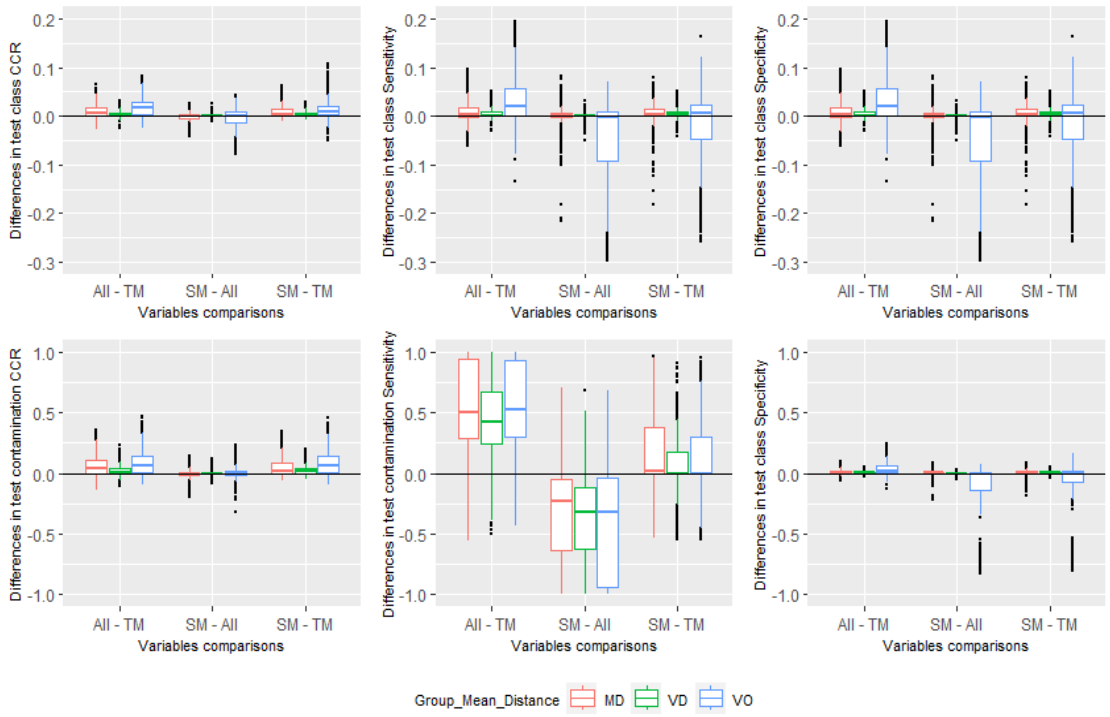



Figure 3.5: Boxplots of test CCR differences for models with different variable sets with different levels of distances between mean classes (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

Table 3.6 displays the differences in the correct classification rate and sensitivity (contamination) for the variable sets with contaminated Gaussian mixture models used to predict the test data. Overall, utilising all variables yields the highest correct classification rate and sensitivity across all performance metrics. However, it is unnecessary to employ all variables to achieve the same classification rate; the variables identified by the greedy search algorithm yield comparable results to using all variables. Conversely, a model constructed solely with true variables exhibits slightly lower levels of correct classification rate and sensitivity.

Table 3.6: Mean performance measures on test datasets for all sets of variables in scenarios with very overlapping distances between classes while other factors were varied.

| Set of Variables | Median CCR (Class) | Median Sensitivity |
|---|---|---|
| True variables | 0.88 | 0.31 |
| Selected variables | 0.90 | 0.58 |
| All variables | 0.90 | 1 |

### 3.3.2 Scenarios with fixed number of classes factor and other factors varied

The performance of the model in predicting classes and identifying contaminated observations was observed by keeping the factor number of classes fixed at 2 and 3 while varying other factors such as the distance between mean classes, number of classes, class proportion, number of variables, percentage of samples used as training, correlation structure, and number of separating variables at each of their levels. Different means of the respective subsets of variables were then taken for test class correct classification accuracy, test class sensitivity, test class specificity, test contamination correct classification accuracy, test contamination sensitivity, and test contamination specificity.

In Figure 3.6, differences in the test class correct classification rate, test class sensitivity, and test class specificity are observed, with the "selected variables" slightly outperforming the "true variables" (since the differences *SM-TM* are positive), especially for two classes. However, this improvement diminishes as the number of classes increases. When comparing the performance of correctly assigning a class to new observations, the subset of "selected variables" is slightly better than the subset of the "all variables" when the number of classes is three, but not when it is two. When comparing the performance of the three sets of variables in identifying contaminated observations, a similar pattern is observed, with the set of the "selected variables" outperforming the set of "true variables" in test contamination classification rate and test contamination sensitivity, although this effect is reduced when a third class is incorporated. The "all variables" subset outperforms the "selected variables" subset in test contamination sensitivity. Additionally, there were no big differences in test contamination specificity among these three sets of variables. The extreme values in the plot may represent cases where the classes were unbalanced, and

one or two of the models failed to produce a good predictive model capable of assigning a class to new observations and identifying contamination.

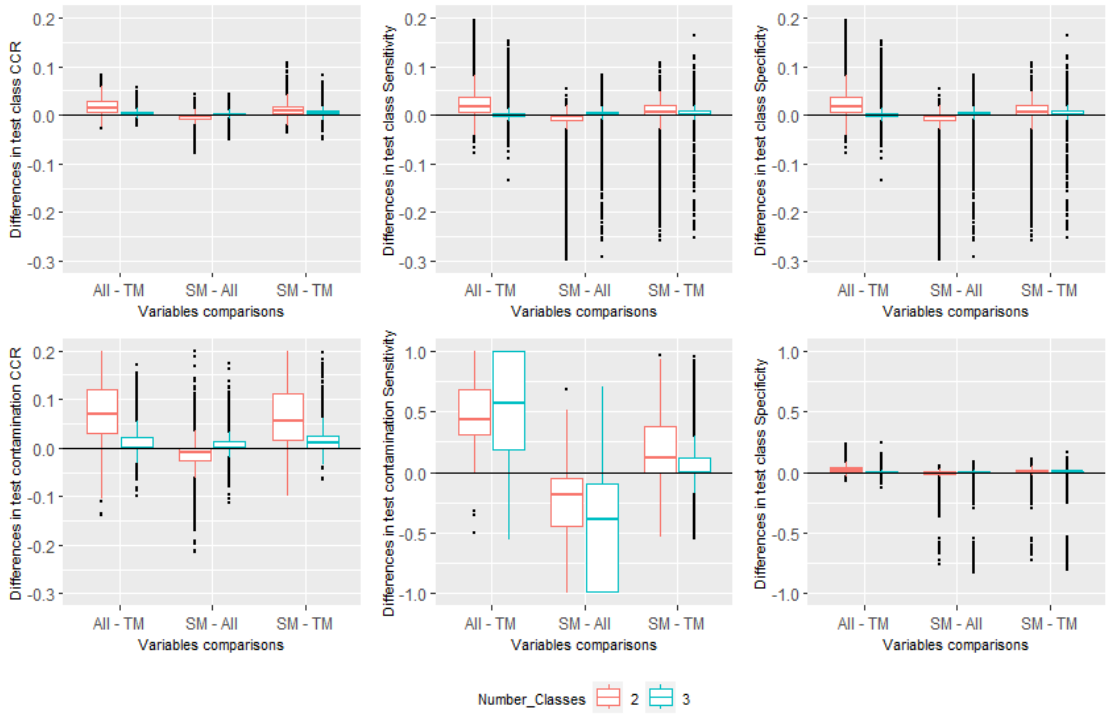



Figure 3.6: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of the number of classes (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.3 Scenarios with fixed class proportion factor and other factors allowed to vary varied

As mentioned earlier, some extreme values were attributed to unbalanced datasets. Figure 3.7 illustrates that the true variables yielded poor class and contamination predictions in certain simulations. It appears that the "selected variables" subset occasionally encounters similar issues. Conversely, the variables incorporating all variables demonstrates superior performance in assigning a class to new observations, as indicated by positive differences *ALL-TM* in test class correct classification rate, test class sensitivity, and test class specificity. However, the "selected variables" only surpasses the "true variables" when classes are balanced, as evidenced by the positive difference *SM-TM*; they perform similarly when classes are unbalanced. When comparing the set of variables in identifying contaminated

observations, it is evident that the "selected variables" is adversely affected by unbalanced classes, resulting in lower levels of test contamination sensitivity and test contamination specificity, particularly when the differences *SM-ALL* are negative. Nonetheless, the "selected variables" subset still outperforms the "true variables" subset in test contamination sensitivity for balanced data *SM-TM*.

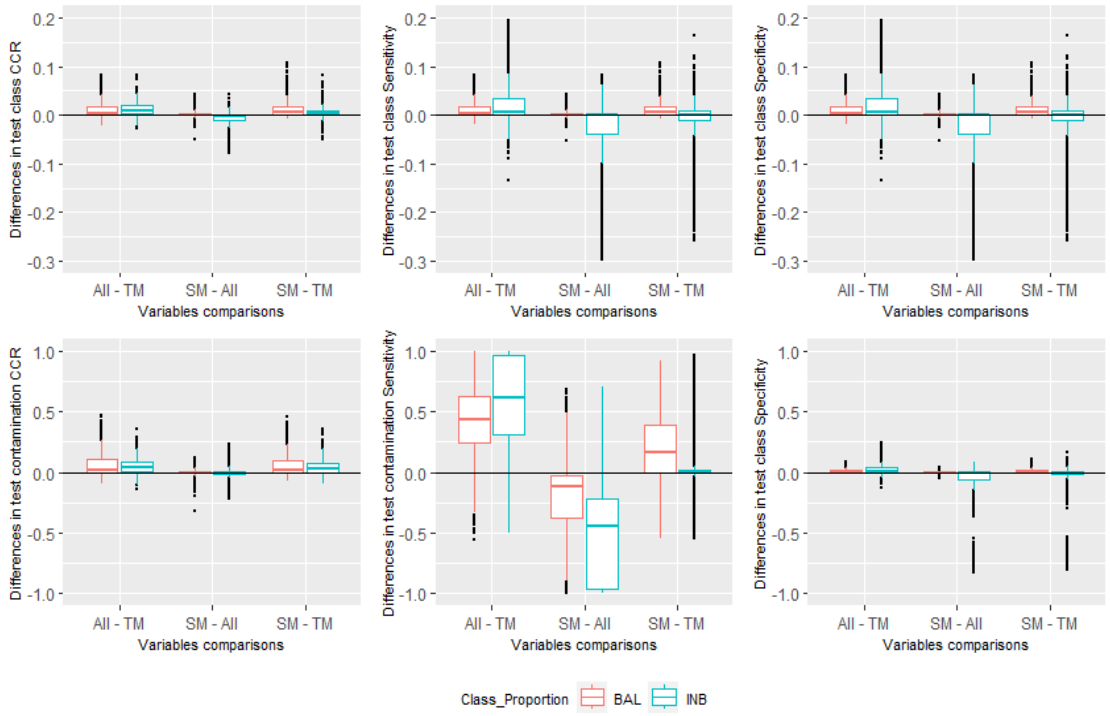



Figure 3.7: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of class proportion (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.4 Scenarios with fixed number of variables factor and other factors varied

In Figure 3.8, when the number of variables was held constant while other factors were varied, marginal differences were observed between the "selected variables" subset and the "all variables" subset in terms of correct classification rate, sensitivity, and specificity within the test class, regardless of the number of variables. Furthermore, both the selected set of variables and the set incorporating all variables exhibited superior performance compared to the true variables in terms of correct classification rate, sensitivity, and specificity

within the test class, as indicated by positive differences in *SM-TM* and *ALL-TM*. When evaluating the performance of the models in identifying contaminated observations, it is observed that the "selected variables" subset and "all variables" subset exhibit superior performance in test contamination correct classification rate compared to the true model. While the "selected variables" subset demonstrates better test contamination sensitivity than the "true variables" subset, its sensitivity is lower compared to the "all variables" subset, especially when the number of variables is 100. This discrepancy arises from the contamination pattern affecting all variables through the inflation factor, resulting in a scenario where some variables lack classification information. Nonetheless, such information could prove useful in identifying contamination.

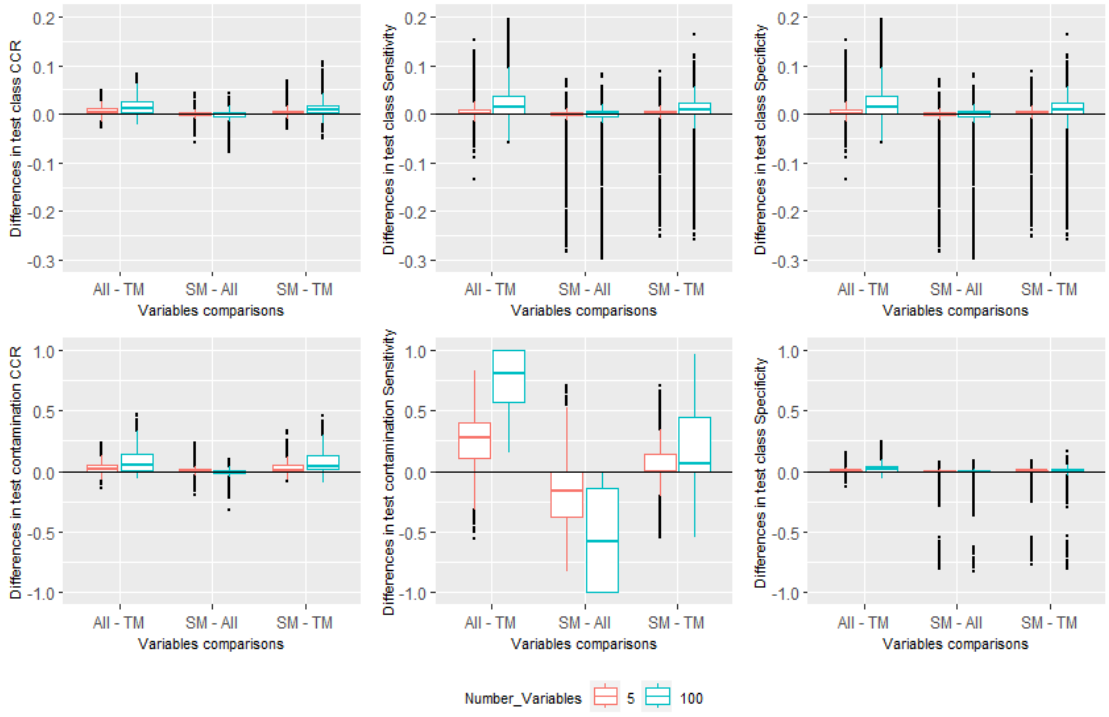



Figure 3.8: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of number of variables (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.5 Scenarios with fixed percentage of samples used in training factor and other factors varied

An evaluation of the performance of the three set of variables (true, selected, all) in classifying and identifying contamination is undertaken. The test class correct classification rate is higher for either the "all variables" subset or the "selected variables" subset, as opposed to the "true variables" subset. Additionally, the direct impact of the percentage of observations utilized in training is not readily apparent. For instance, in terms of test class sensitivity, it is observed that the median of the differences and the interquartile range are positive. However, there are some observations deemed as outliers, arising from scenarios primarily characterized by unbalanced data, strongly correlated variables, or a combination of both. In essence, having a larger number of observations in the training set may contribute to improving the training stage of the model. Nevertheless, in this scenario, there is no clear indication that an increase of 10% in the number of observations in the training set correlates with improved performance metrics of the models.

Figure 3.9 shows the performance of the three set of variables when the percentage of samples used in training was held constant for each of its levels while the other factors were varied. The test class correct classification rate shows that although there are no differences between the "selected variables" subset and the "all variables" subset, both of these sets of variables outperform the "true variables" subset regardless the level of proportion of observation assigned to the training set. The same pattern is visible in test class sensitivity and test class specificity. The variation of the differences in the test class correct classification rate is smaller than the differences in the test class sensitivity and test class specificity. In terms of contamination performance there are not big differences between using the "selected variables" and "all variables" subsets. The use of the subset "selected variables" and "all variables" subsets outperform the use of "all variables" subset. The differences in test contamination sensitivity between using the "selected variables", "all variables" subsets and "true variables" are positive which means that it is better to use the first two both subsets of variables, but the use of the "all variables" subset produces better identification of contaminated samples than using the "selected variables" subset regardless the level of the proportion of observation assigned to the training set. Although there differences in test classification specificity between the "selected variables" and "all variables" are very small, there is a slight better performance in specificity using the "selected variables" and "all variables" subsets than the "true variables" subset. Ad-

ditionally, it is possible to observe lower differences between "selected variables" and "all variables" in test class sensitivity, test class specificity, test contamination sensitivity, test contamination specificity which occur in scenarios where the "selected variables" subset produced poor class predictions, especially in unbalanced data.

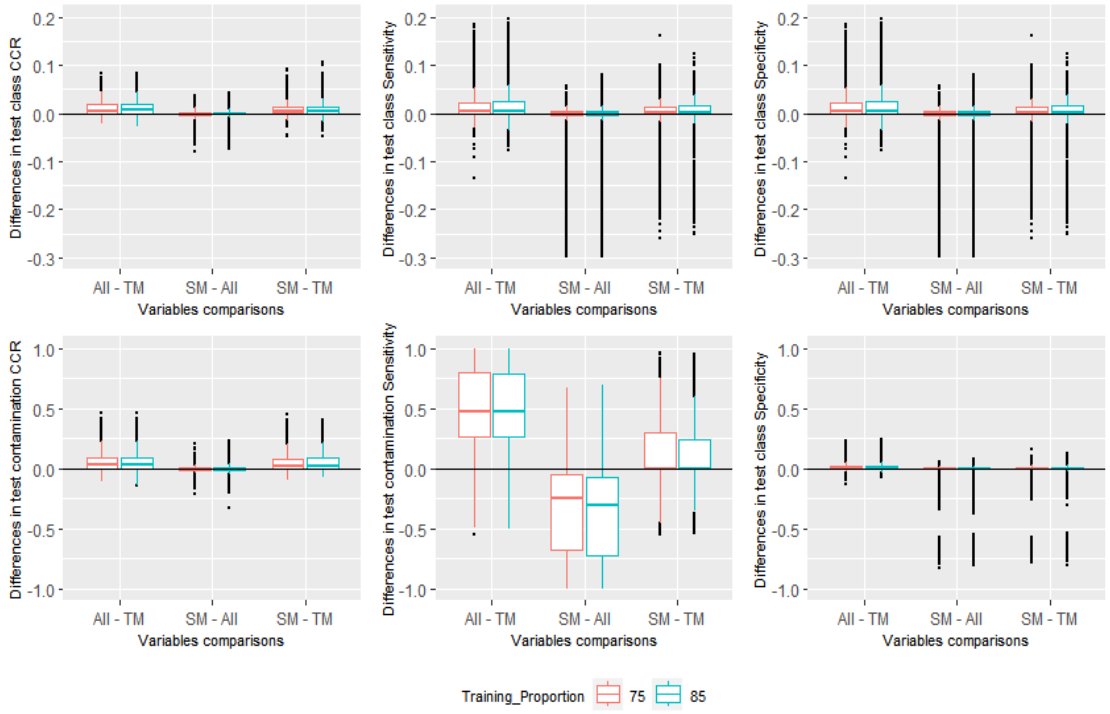



Figure 3.9: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of proportion of observations for training (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.6 Scenarios with fixed correlation structure factor and other factors varied

In Figure 3.10 the differences in test class correct classification rates are positive when comparing "selected variables", "all variables", and "true variables". However, there is not much difference in the test class correct classification rates and test contamination correct classification rates between "selected variables" and "all variables" regardless the type of correlation structure. Assessing the behaviour of the effect of the correlation type on the ability of the model to identify contaminated samples, it is clear that the correlation structure with the lowest sensitivity is a strong correlation between separating variables. The

"selected variables" and "all variables" subsets perform better than the "true variables" subset in test contamination correct classification rate and test contamination sensitivity. Additionally, including the "all variables" subset outperforms including only the "selected variables" subset in identifying correctly contaminated samples as the test contamination sensitivity metric shows. Moreover, the model including all the variables also performs better in terms of specificity which identifies correctly non-contaminated observations and there is not a big difference between the specificity yields by the "selected variables" and "true variables" subsets.

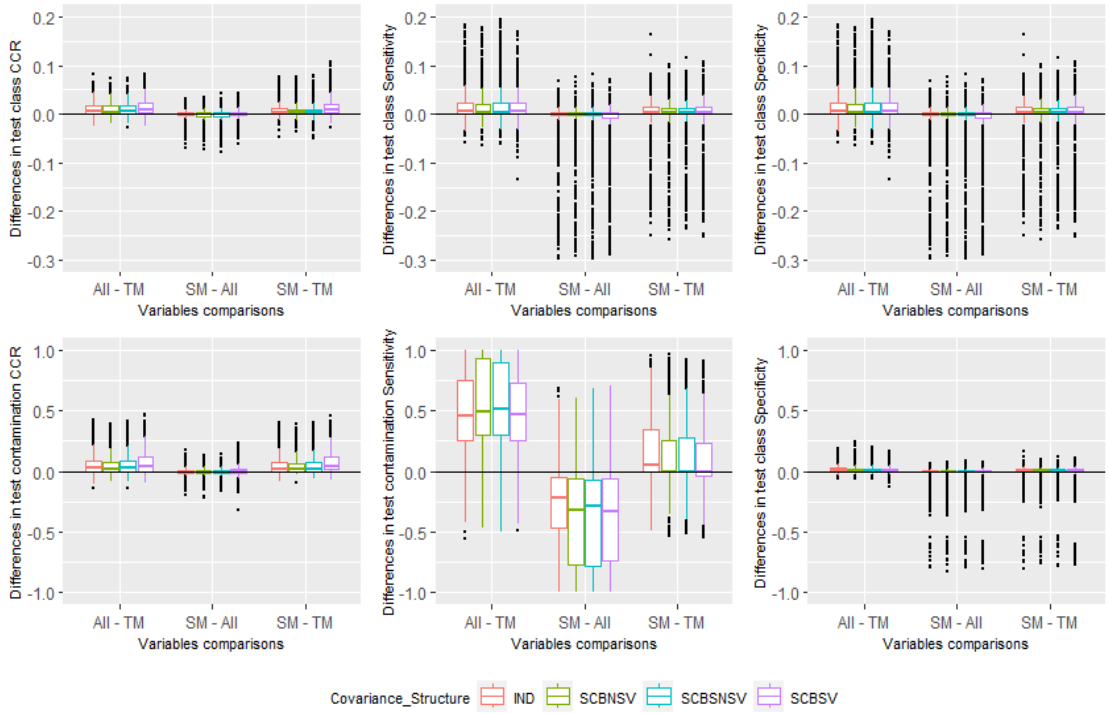



Figure 3.10: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of correlation structure fixed and other factors varied (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.7 Scenarios with factor number of separating variables fixed and other factors varied

The differences in performance metrics for the three different sets of variables, when the number of separating variables is fixed while the other factors are varied, are shown in Figure 3.11. In the test class correct classification rate there is not much difference between

the “selected variables” and the “all the variables” subsets. Nevertheless, the “selected variables” and “all variables” subsets performed better in terms of the test class correct classification rate than the “true variables” subset regardless the number of separating variables. Looking at the test class sensitivity and test class specificity the “true variables“ and the “selected variables” subsets yield sometimes lower classification mainly when there are unbalanced classes. Looking at the performance metrics that measure the ability of models to identify contaminated samples, both sets of variables the selected and the set of all the variables produce similar test class correct classification rates and outperform the true variables in this metric. Moreover, the selected variables and the set of all the variables outperform the true variables in test contamination sensitivity and the set of all the variables yields higher test contamination sensitivity than the model formed by the selected variables. In terms of variability of the performance metrics, it is visible that test class and test contamination correct classification rate have lower variability, while test class and test contamination sensitivity and specificity have higher variability. The reasons behind this is that there were cases where the “selected variables” and “true variables” subsets were either not able to detect an important number of observations that belong to a class or were contaminated. Consequently, this might leads to cases where computing either sensitivity or specificity produces an in-determination that is replaced by zero. Hence, this is reflected in higher negative values around 0.5 units in the plot of differences. The reason behind this is that there were scenarios where either the “selected variables” or “true variables” subsets did not contain the complete group information nor contamination information.

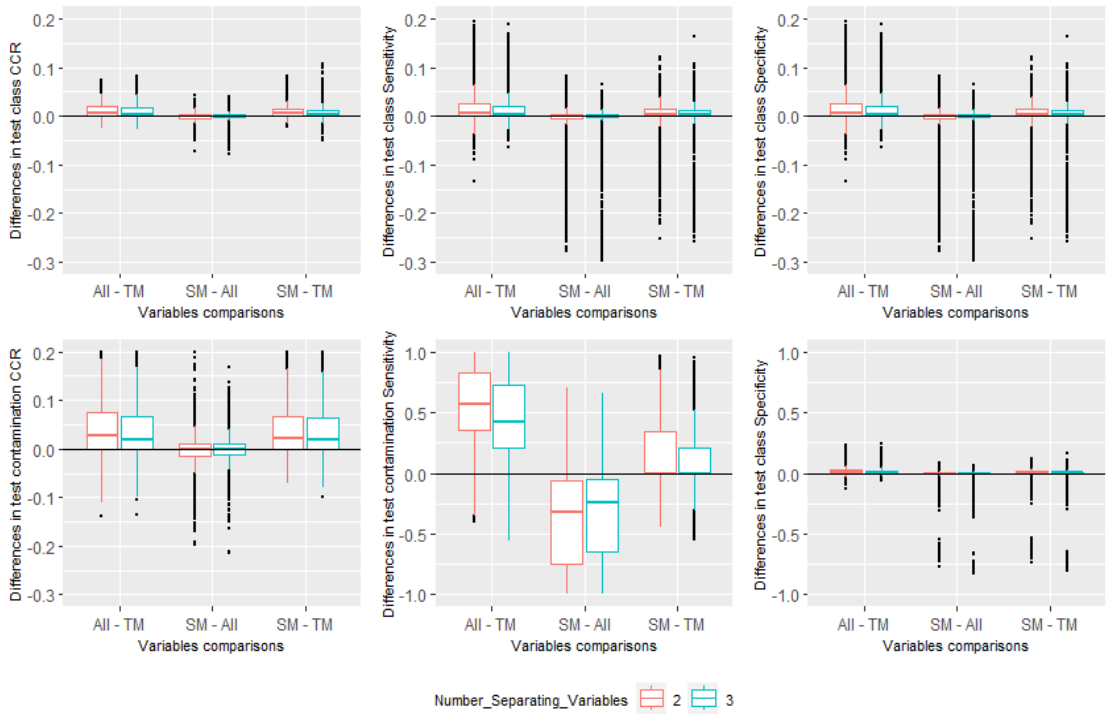



Figure 3.11: Boxplots of test CCR,, sensitivity, and specificity differences for models with different variable sets with different levels of number of separating variables (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 3.3.8 Modeling mean of correct classification rate (CCR) and sensitivity by factors

Upon considering the combined performance of the three sets of variables, it becomes necessary to determine whether the effectiveness of the proposed approach depends on specific circumstances, particularly if performance varies across different factor levels. The performance of each set is assessed concerning the levels of each factor. It is anticipated that there will be fluctuations in performance across different settings due to varying factors at their respective levels. Between-scenario effects can be interpreted as the influence of the design factors across all three sets of variables under consideration.

A regression analysis with repeated measures is conducted, focusing solely on the main effects, which are modelled and discussed in detail. The model's intercept serves as the baseline for correct classification rate and sensitivity, assuming the inclusion of true variables in the model, a moderate distance between classes, balanced class proportions, 2

classes, a small number of independent variables, and two separating variables.

In Table 3.7, the distance between classes, the subset of variables selected, class proportion, number of separating variables, and the covariance structure emerge as the most influential factors on the test class correct classification rate. The subset of variables selected, number of variables, and class proportion emerge as the most influential factors on the test contamination sensitivity.

In Table 3.8, a highly overlapping distance has a negative impact, while a large separation between classes has a positive effect. Moreover, a strong correlation between separating variables impairs the model. Additionally, including either the selected or all variables improves the model's ability to allocate new observations to their corresponding classes. All other factors are significant except for the number of classes. The highest positive impact on sensitivity is observed when either variables are obtained by Greedy search or all variables are included in the model. Furthermore, having an additional separating variable positively affects sensitivity. Conversely, the most detrimental scenario for sensitivity occurs when classes are imbalanced and separating variables are strongly correlated.

Table 3.7: Analysis of variance for the three sets of variables on correct classification rate

| Source | Df | F (CCR) | F (Sensitivity) | p-value (CCR) | p-value (Sensitivity) |
|---|---|---|---|---|---|
| Intercept | 1 | 6383505 | 63986.25 | $< .0001$ | $< .0001$ |
| Distance between classes | 2 | 8805 | 9.87 | $< .0001$ | $< .0001$ |
| Variables | 2 | 2446 | 18096.10 | $< .0001$ | $< .0001$ |
| Class proportion | 1 | 1408 | 1901.96 | $< .0001$ | $< .0001$ |
| Number of separating variables | 1 | 548 | 48.39 | $< .0001$ | $< .0001$ |
| Covariance structure | 3 | 251 | 34.16 | $< .0001$ | $< .0001$ |
| Number of variables | 1 | 32 | 2883.70 | $< .0001$ | $< .0001$ |
| Number of classes | 1 | 6 | 271.56 | $< .0139$ | $< .0001$ |
| Number of classes:Variables | 2 | 826 | 96.36 | $< .0001$ | $< .0001$ |
| Number of variables:Variables | 2 | 495 | 2579.55 | $< .0001$ | $< .0001$ |
| Class proportion:Variables | 2 | 350 | 1901.96 | $< .0001$ | $< .0001$ |
| Distance between classes:Variables | 4 | 237 | 9.87 | $< .0001$ | $< .0001$ |
| Number of separating variables:Variables | 2 | 23 | 48.39 | $< .0001$ | $< .0001$ |
| Covariance structure:Variables | 6 | 21 | 19.06 | $< .0001$ | $< .0001$ |

Table 3.8: Coefficient estimates of the model with interactions to explain CCR

| Sources | Estimates (CCR) | t value (CCR) | Estimates (Sensitivity) | t value (Sensitivity) |
|---|---|---|---|---|
| Intercept | 0.89 | 385.19 | 0.07 | 3.74 |
| Number of Classes - 3 | 0.00 | 0.68 | 0.05 | 7.45 |
| Variables All | 0.02 | 14.62 | 0.59 | 39.15 |
| Variables Selected | 0.02 | 14.70 | 0.54 | 23.85 |
| Number of separating variables | 0.02 | 26.08 | 0.12 | 19.93 |
| Number of variables - 100 | -0.00 | -0.58 | 0.04 | 6.11 |
| Distance between classes- VD | 0.03 | 33.35 | 0.07 | 8.76 |
| Distance between classes - VO | -0.09 | -95.61 | -0.01 | -1.30 |
| Class proportion - INB | 0.03 | 38.22 | -0.25 | -37.36 |
| Covariance Structure - SCBNSV | 0.00 | 3.22 | -0.06 | -6.31 |
| Covariance Structure - SCBSNSV | 0.00 | 2.92 | -0.03 | -6.92 |
| Covariance Structure - SCBSV | -0.03 | -22.20 | -0.03 | -3.25 |
| Variables All*Number of classes - 3 | -0.00 | -35.24 | -0.03 | -5.71 |
| VariablesSelected*Number of classes 3 | -0.01 | -15.95 | -0.10 | -11.42 |
| Variables All*Number of separating variables - 3 | -0.00 | -4.45 | -0.14 | -26.52 |
| VariablesSelected*Number of separating variables -3 | -0.00 | -6.71 | -0.09 | -11.48 |
| Variables All*Number of variables - 100 | 0.01 | 29.43 | 0.36 | 70.32 |
| VariablesSelected*Number of variables - 100 | 0.01 | 23.66 | 0.17 | 21.06 |
| VariablesAll* Distance between classes-VD | -0.01 | -14.57 | -0.06 | -10.11 |
| VariablesSelected*Distance between classes-VD | -0.01 | -10.95 | -0.14 | -14.53 |
| VariablesAll*Distance between classes-VO | 0.01 | 16.17 | 0.01 | 2.07 |
| /VariablesSelected*Distance between classes-VO | 0.00 | 6.96 | 0.01 | -9.31 |
| VariablesAll*Proportion - INB | 0.00 | 3.09 | 0.16 | 30.72 |
| VariablesSelected*Proportion - INB | -0.01 | -18.75 | -0.15 | -19.28 |
| Variables All*CovarianceStructure-SCBNSV | -0.00 | -3.15 | 0.04 | 5.07 |
| Variables Selected*CovarianceStructure-SCBNSV | -0.00 | -2.41 | -0.05 | -4.6 |
| VariablesAll*CovarianceStructure-SCBSNSV | -0.00 | -2.72 | 0.04 | 5.00 |
| VariablesSelected*CovarianceStructure-SCBSNSV | -0.01 | -2.89 | -0.03 | -3.20 |
| VariablesAll*CovarianceStructure-SCBSV | 0.00 | 5.32 | 0.01 | 0.68 |
| VariablesSelected*CovarianceStructure-SCBSV | 0.00 | 6.81 | -0.09 | -7.72 |

The interactions between the set of variables and the other factors were found to be significant, yet their influence on the correct classification rate appears to be minor. The interaction between increasing the number of variables and employing a variable selection procedure or including all variables in the model has a positive impact on sensitivity. Moreover, the interaction between the set of variables used and having unbalanced classes results in a decrease in sensitivity. This suggests a potential shortage of observations in one of the classes, affecting the classification model's ability to identify contaminated samples.

### 3.3.9 Inclusion of informative and non-informative variables

As stated earlier, alongside achieving good classification performance, a desirable characteristic in a variable selection algorithm lies in its capability to distinguish informative

variables from non-informative ones. To explore this aspect further, the proportion of simulations in which each informative variable is individually chosen into the model by the greedy search algorithm is documented for instances where the number of informative variables was 2 and 3. Additionally, the overall proportion of informative variables incorporated into the model, along with the proportion of non-informative variables that are excluded from the selected model – referred to as inclusion correctness and exclusion correctness, respectively – is monitored.

Table 3.9 depicts the behaviour of variable selection in scenarios featuring two separate variables. The first row presents a summary of instances where the number of available variables was 5. On average, 3 variables were selected to form the model, with separating variables $X_2$ and $X_4$ appearing in 92% and 91% of cases within the subset of selected variables, respectively. The variable selection procedure, on average, included either one or both of the separating variables 91% of the time, while discarding, on average, 61% of non-informative variables during the variable search. The second row presents a summary of instances where the number of available variables was 100. On average, 6.98 variables were selected to form the model, with separating variables $X_2$ and $X_4$ appearing in 84% and 82% of cases within the subset of selected variables, respectively. The greedy search algorithm, on average, included either one or both of the separating variables 83% of the time, while discarding, on average, 95% of non-informative variables during the variable search.

Table 3.9: Variation in the inclusion of separating variables and exclusion of non-informative ones in the scenario of two separating variables by the greedy search algorithm across varied factor levels

| Number of separating variables | Number of variables | Average number of selected variables variables | $X_2$ | $X_4$ | Inclusion correctness | Exclusion correctness |
|---|---|---|---|---|---|---|
| 2 | 5 | 3.00 | 92% | 91% | 91% | 61% |
| 2 | 100 | 6.98 | 84% | 82% | 83% | 95% |

The boxplot in Figure 3.15 illustrates the distribution of the number of variables in-

cluded in the "selected variables" subset. The results indicate that, on average, 3 variables were included in the model for classes mapped in 5 dimensions, while 6 variables were included for classes mapped in 100 dimensions. In terms of variability in the number of variables included, there is greater variability when classes are mapped in 100 dimensions, attributable to a larger number of subsets from which to choose. Additionally, for scenarios where classes were mapped in 5 dimensions, the number of variables included varied within the range $[1-3]$, with one instance where all 5 variables were selected. Conversely, for scenarios where classes were mapped in 100 dimensions, the number of variables included varied within the range $[3-8]$, with occasional instances where between $15-20$ variables were selected.

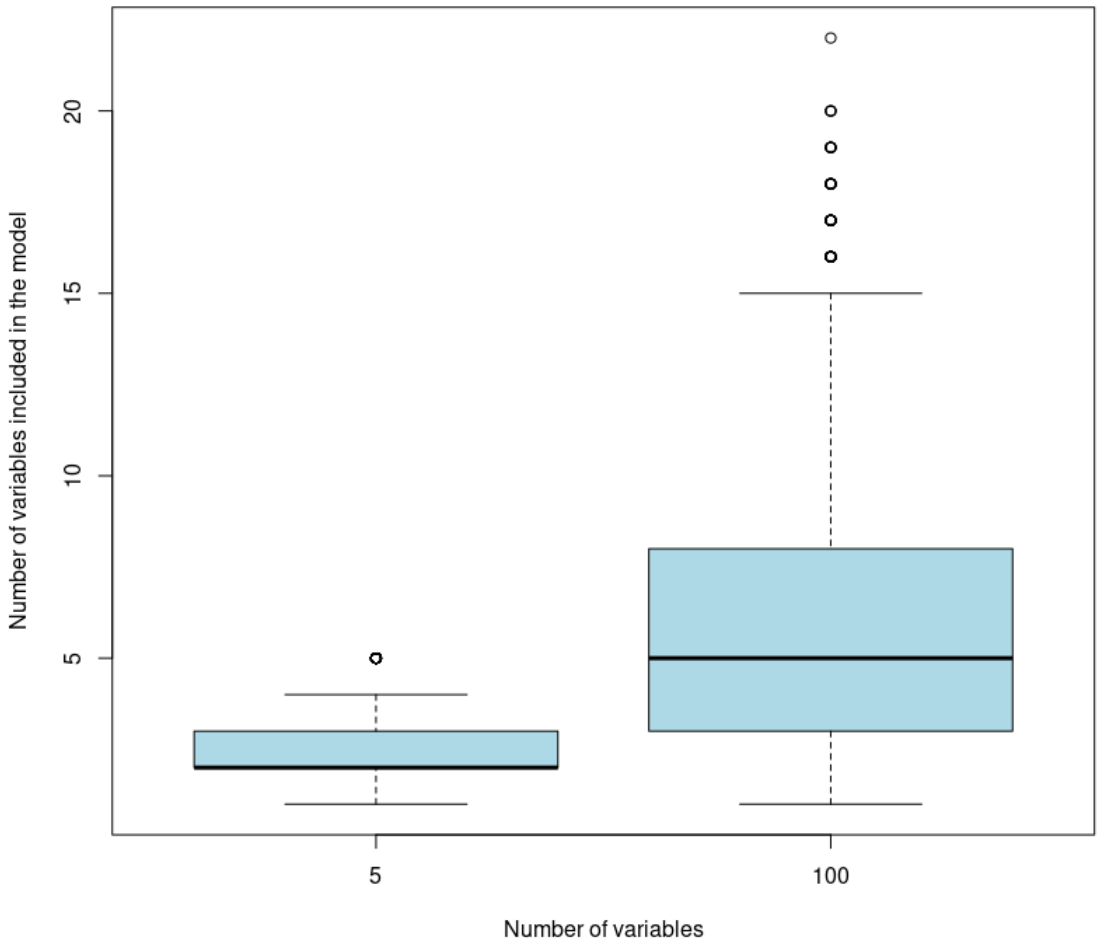



Figure 3.12: Variation of the number of variables selected by the greedy search algorithm across simulated datasets with two separating variables

The distribution of inclusion correctness is depicted in Figure 3.13. In scenarios involving two separating variables, it is observed that the true variables are predominantly identified and included, with occasional instances where one or more of the true separating variables are not included by the variable search procedure. In scenarios where the number of variables was 100, the separating variables were identified by the greedy search algorithm between 60% and 100% of the time, with occasional instances where none of the true variables could be included in the model. The cases where the variable search procedure failed to identify any of the separating variables into the selected model had a common characteristic: they were scenarios with unbalanced classes.

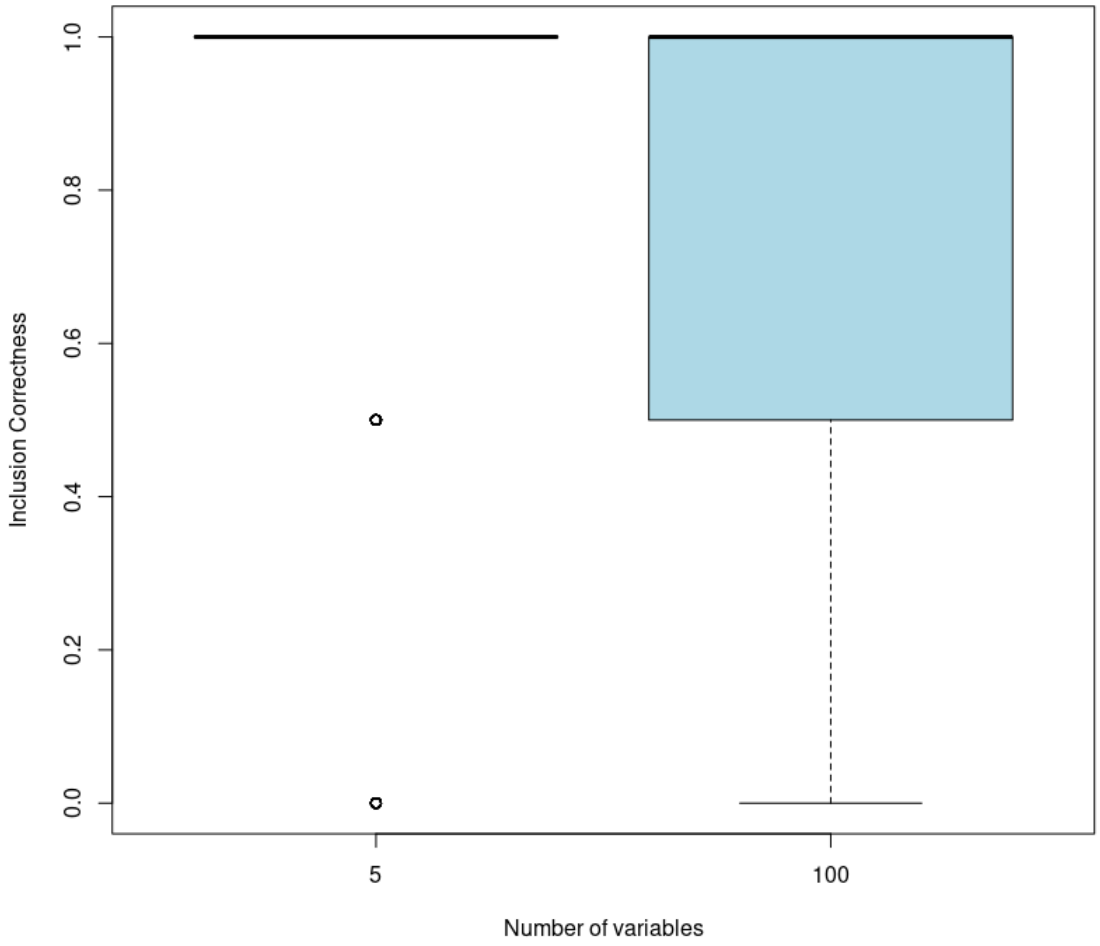



Figure 3.13: Variation of the inclusion correctness for scenarios with two separating variables

The exclusion correctness distribution is portrayed in Figure 3.14. In scenarios where

two separating variables are mapped in five dimensions, the exclusion correctness reveals that the variable search excludes over 67% of non-separating variables, with a few exceptional cases where it excludes less than 40% of non-separating variables. These exceptional cases were only present in scenarios with classes mapped in 5 dimensions. In cases where two classes were mapped in 100 dimensions, the variable search algorithm managed to discard between 94% and 99% of the non-informative variables, with a few exceptional cases where it discarded just under 90% of non-separating variables. The exceptional cases where the variable search did exclude less than 40% of the non-separating variables were mainly in scenarios with unbalanced classes where there were a small number of observations in one of the classes.

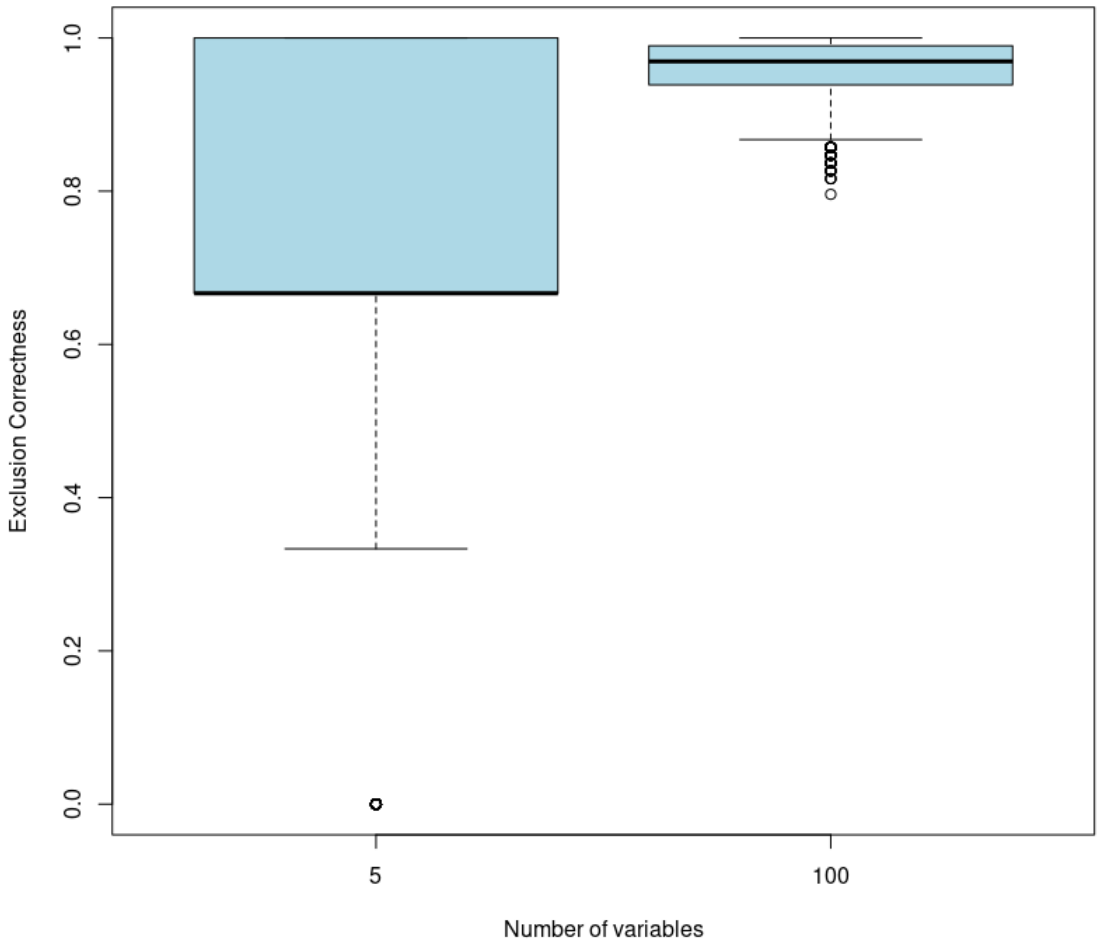



Figure 3.14: Variation of the exclusion correctness for scenarios with two separating variables

In Table 3.10 a summary of some metrics to assess the greedy search algorithm is presented. The average number of variables selected by the greedy search algorithm in scenarios where the classes were mapped in 5 and 100 dimensions were 3 and 7 variables respectively. Moreover, the greedy search algorithm in cases with three separating variables, $X_2$, $X_4$, and $X_5$ were included in $87\%, 83\%$, and $94\%$ of the models, respectively for classes mapped in 5 dimensions. For classes mapped in 100 dimensions, the three separating variables were included in the model 82%, 81%, and 85% of times respectively. Evaluating the ability of the greedy search to include separating variables and exclude the non-separating variables, it is possible to see that for classes mapped in 5 dimensions, the variable search procedure includes 88% of the separating variables and excludes 61% of non-separating variables. In scenarios when classes are mapped in 100 dimensions, the algorithm search included 82% of separating variables while excluding 95% of non-separating variables. The results suggest that when the number of non-informative variables increases, the performance of the greedy search algorithm in discarding the non-informative variables improves.

Table 3.10: Variation in the inclusion of separating variables and exclusion of non-informative ones in the scenario of three separating variables by the greedy search algorithm across varied factor levels

| Number of separating variables | Number of variables | Number of selected variables | $X_2$ | $X_4$ | $X_5$ | Inclusion correctness | Exclusion correctness |
|---|---|---|---|---|---|---|---|
| 3 | 5 | 3.41 | 87% | 83% | 94% | 88% | 61% |
| 3 | 100 | 7.10 | 82% | 81% | 85% | 83% | 95% |

In Figure 3.15 the distribution of the number of variables included in the model is observed. It shows that in cases where the number of variables was small (5 variables), the average number of variables included in the model varied between $3 - 4$ with some cases with only one variable included. Additionally, the number of variables included in the model when the classes were mapped in 100 dimensions was between 3 and 9 occasionally including between 15 and 20 variables when these variables somehow added a marginal improvement to the test class correct classification rate.

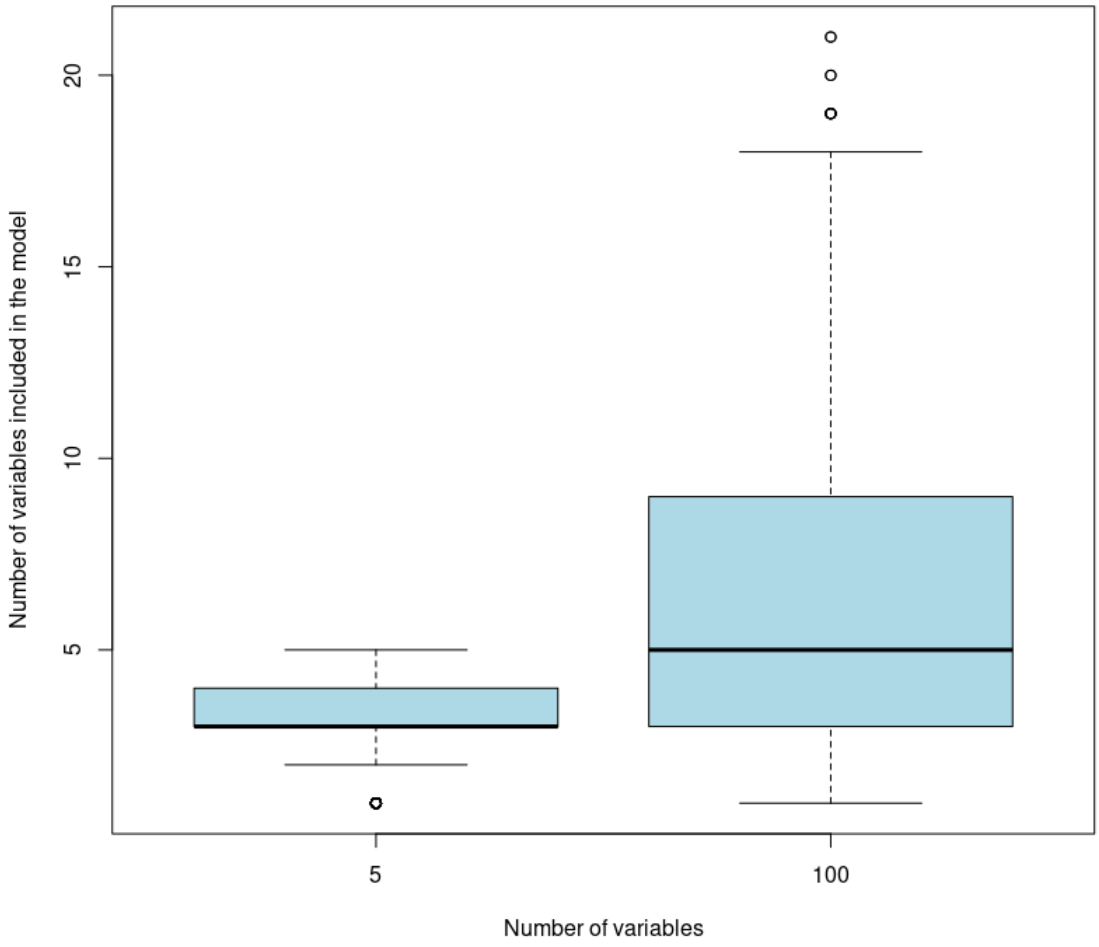



Figure 3.15: Variation of the number of variables selected by the greedy search algorithm across simulated datasets with tree separating variables

The distribution of inclusion correctness, as depicted in Figure 3.16, indicates that for classes mapped in 5 dimensions, the vast majority of the time, all the separating variables were included by the greedy search algorithm in the selected model. In very few cases, only one of the separating variables or none of them were included. In scenarios where the classes were mapped in 100 dimensions, the range of separating variables included in the "selected variables" subset varied. In most instances, all of them were identified and included, but occasionally only half were identified and included. There were exceptional cases, mainly scenarios with unbalanced classes, where the variable search failed to identify any of the separating variables.

/

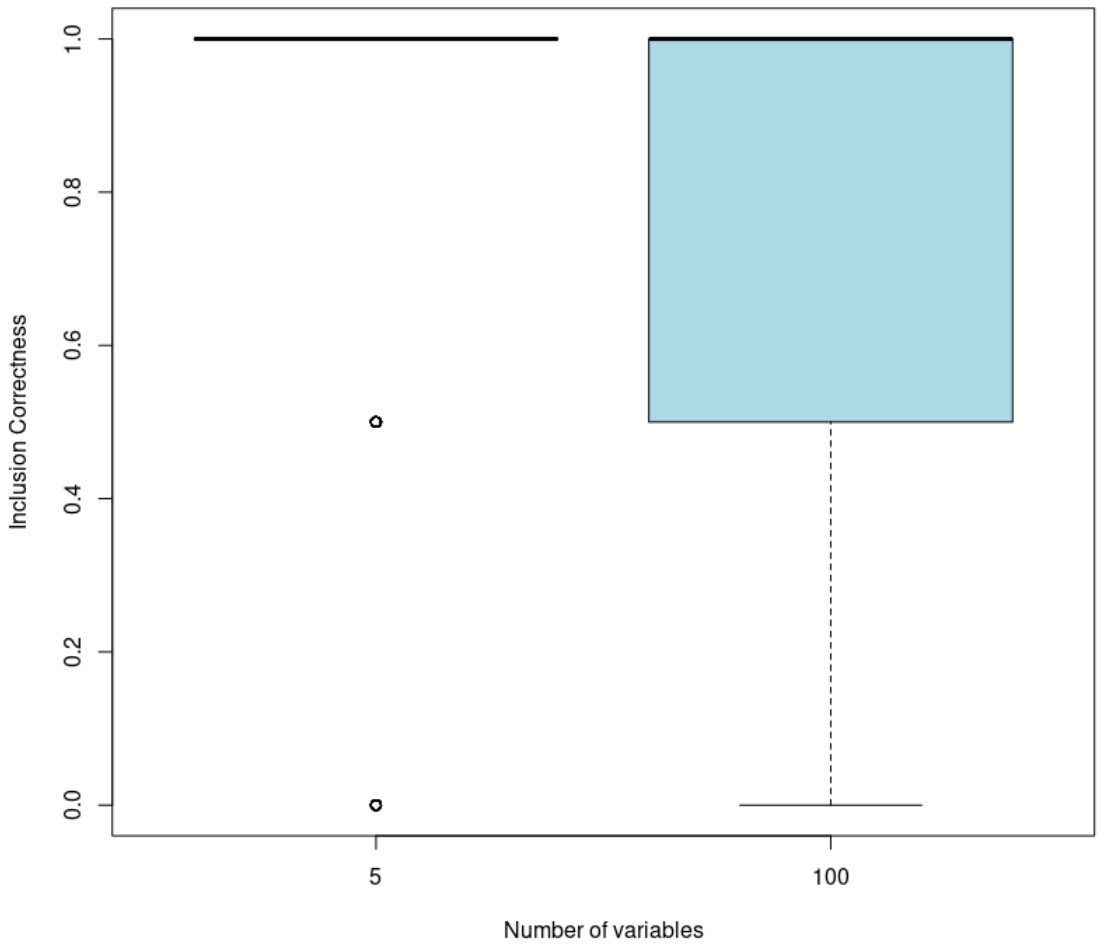



Figure 3.16: Variation of the inclusion correctness for scenarios with three separating variables

In Figure 3.17, the distribution of exclusion correctness is depicted for classes mapped in 5 and 100 dimensions. In scenarios where the classes were mapped in 5 dimensions, the variable search procedure predominantly excluded more than 60% of non-separating variables, with very few instances where all variables were included in the "selected variables" subset. These few scenarios shared a common characteristic: unbalanced classes in the dataset. In scenarios where classes were mapped in 100 dimensions, the variable search procedure discarded more than 90% of the non-separating variables and occasionally discarded between 80% to 90% of the non-separating variables in scenarios with unbalanced classes.

/

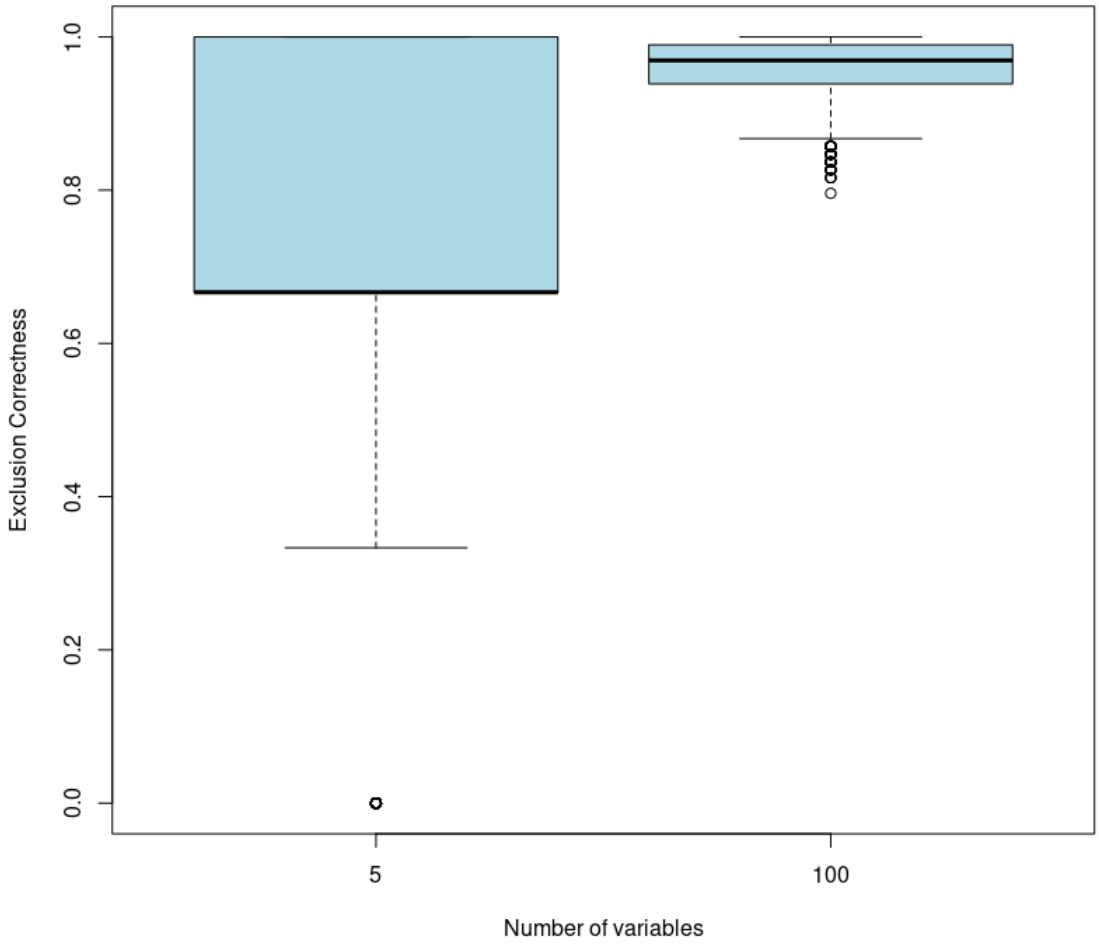



Figure 3.17: Variation of the exclusion correctness for scenarios with three separating variables

In summary, models generated by the greedy search algorithm typically consist of a smaller subset of variables, which includes the separating variables among them. Occasionally, fluctuations occur, resulting in the addition of more non-separating variables to the model, aiming to compensate for the lack of observations in scenarios with unbalanced classes.

## 3.4 Plasmode data sets

The results of the simulation study show promise, although there were some concerns, particularly in cases where the dataset is imbalanced or when the variables used for separation

are strongly correlated. The disadvantage of using simulated data sets is that simulated data rarely captures the many complexities that occur in real data. For example, non-normality, non-linearity, high-level correlations, etc. routinely happen in real applications but are hard to reproduce in simulations. On the other hand, if the real data does not have a known truth about an aspect the developed methodology is trying to test for, then it has a disadvantage over simulated data which can be constructed to have that known truth.

This section will use plasmode data to attempt to overcome these issues. Plasmode data (Gadbury et al., 2008) is data that is derived from datasets that occur in reality. It will have the inherent complexity associated with real data but has been augmented to allow some particular truth to be known within the resulting data. Three sets of plasmode datasets are presented based on the following real data: Blue Crabs (Campbell and Mahon, 1974), Wine Forina et al. (1988), Dua and Graff (2019) and Breast Cancer datasets .

The aim is to compare the performance of a contaminated mixture Gaussian model employing variable selection against a mixture of contaminated Gaussians incorporating all variables in datasets with high correlation (and other potential complexities) while varying the parameters that control the level of contamination. This comparison is intended to classify new observations and determine whether they are contaminated. The evaluation involves assessing the classification accuracy and the ability to identify contaminated samples by examining the differences in correct classification rates and sensitivities between models using selected variables and those using all variables. A positive difference indicates that the model with selected variables outperforms the full model, while a negative difference suggests the opposite. Because this is based on real data rather than simulations, in this case, there is no ground truth in terms of which variables are class separating.

## 3.5 Crab data

The crab dataset (Campbell and Mahon, 1974), comprises 200 samples, with equal representation of 50 males and 50 females for both the blue and orange species of rock crab. In each specimen there are measurements for various parameters, including (1) front global size (FL), (2) posterior region width (RW), (3) carapace length (CL), (4) carapace width (CW), and (5) body depth (BD). All measurements in millimeters. Among others, this

dataset was analyzed by Dean (2006), Ripley (2007), Hennig (2010), and Yan (2017). Here, the number of classes and observations' class memberships are known.

Peel and McLachlan (2000) notice that assuming multivariate normality with a common covariance matrix is a reasonable assumption for this dataset. In the following analyses only 100 crabs from the blue species will be considered and the classes of interest for classification are sex (male and female). The dataset comprises five variables exhibiting high correlation as Table 3.11 shows. The pairs plot in Figure 3.18 of the original crabs data (before augmentation) reveals distinct separation between the males and females with minimal overlap. Notably when plotting $CL$ against $RW$, a clear distinction between sexes emerges. As was expected, since these are physical measurements all recording some aspect of crab size, the variables form an upward linear trend from left to right when they are plotted in pairs as a result of being highly positively correlated.

Table 3.11: Correlation Matrix for Original Crab Data

| | FL | RW | CL | CW | BD |
|---|---|---|---|---|---|
| FL | 1 | 0.91 | 0.98 | 0.96 | 0.99 |
| RW | 0.91 | 1 | 0.89 | 0.90 | 0.89 |
| CL | 0.98 | 0.89 | 1 | 1 | 0.98 |
| CW | 0.96 | 0.90 | 1 | 1 | 0.97 |
| BD | 0.99 | 0.89 | 0.98 | 0.97 | 1 |

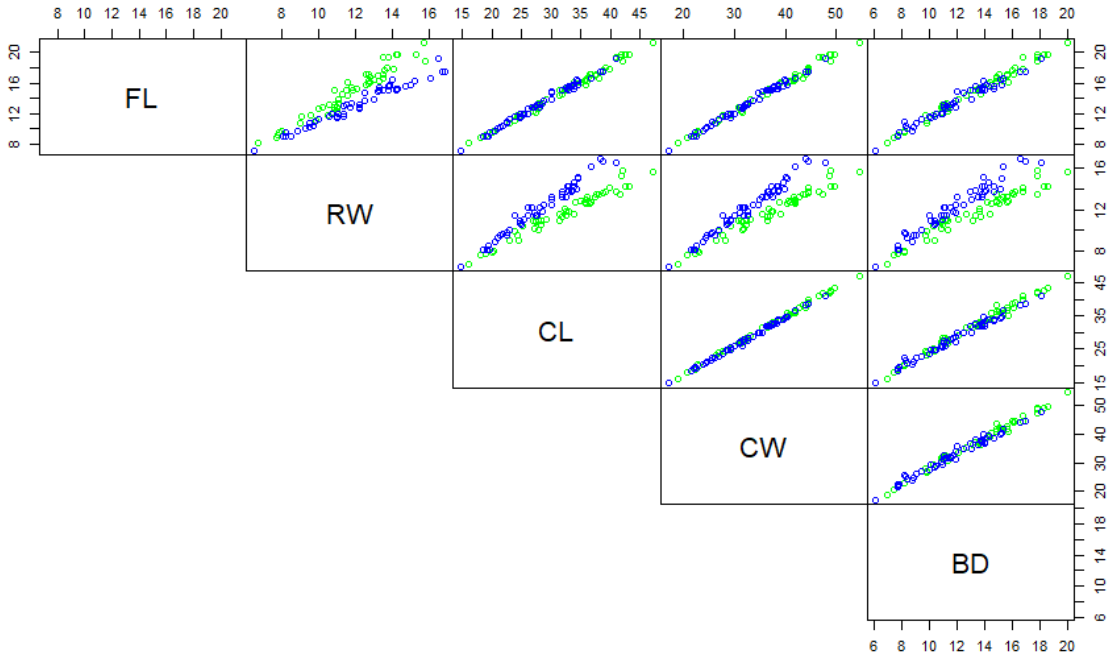



Figure 3.18: Pairs plot depicting the relationships between various features of uncontaminated male and female specimens of blue crabs. Female specimens (F) are shown in green, while Male specimens (M) are shown in blue.

The procedure of how contaminated samples are generated and introduced into the blue crab samples is described in the next section.

### 3.5.1 Contaminating crab data

The *blue crab* subset is balanced, featuring an equal proportion of samples for male and female classes. A fixed percentage of samples, denoted as F5, is used to fit the model, set at 70%. Two factors governing contamination, namely the percentage of non-contaminated samples (F7) and the variance inflation (F8) factor, are assessed at two levels: equal and non-equal, with varying values for both sexes. These specific values are outlined in Table 3.12, resulting in a total of 810 simulations across 81 potential scenarios. Each scenario is repeatedly simulated, generating 10 datasets for each combination.

Table 3.12: Variation of factor levels, non-contaminated samples percentages, and inflation factors across sex and simulated studies of blue crabs

| Factors | Description | Levels |
|---|---|---|
| $F_7$ | Percentage of non-contaminated samples | M (75%, 80%, 85%).<br>F (75%, 80%, 85%). |
| $F_8$ | Variance inflation factor | M (5, 10, 15).<br>F (5, 10, 15 ). |

To generate contaminated samples, the respective sample mean $\hat{\boldsymbol{\mu}}_g$ and covariance $\hat{\boldsymbol{\Sigma}}_g$ are calculated for each of the male and female classes $g = 1, 2$. Subsequently, contaminated samples are generated from a contaminated Gaussian distribution with parameters $N(\hat{\boldsymbol{\mu}}_g, \eta_g\hat{\boldsymbol{\Sigma}}_g)$, where $\alpha_g$ and $\eta_g$ take their corresponding values for each setting (see Table 3.12).

The reason behind varying the percentage of non-contaminated samples for both sexes at levels of $(75\%, 80\%, 85\%)$ is rooted in the observation that real datasets typically exhibit low levels of contamination. Similarly, the inflation factor for both sexes is varied across small ($\eta = 5$), medium ($\eta = 10$), and large ($\eta = 15$). The performance of the model in allocating correctly a new observation into a class an identified whether it is contaminated is assessed for scenarios where $\alpha$ and $\eta$ are equal and unequal across sexes.

As previously noted, the variables $CL$ and $RW$ contribute most to the separation between sexes. Hence, it is of most interest to explore how contaminated samples simulated at different values of $\eta$ affect these variables. This is plotted in Figures 3.19, 3.20 and 3.21. In the visual representation, non-contaminated and contaminated observations are denoted by symbols ($\bullet$ and $\triangle$ respectively), while male and female crabs are distinguished by blue and green colors.

To examine the impact of varying the inflation on the separation of variables, scatter plots for the $CL$ and $RW$ variables are presented for simulations where the percentage of non-contaminated observations is fixed at 75%. In these scatter plots, it becomes apparent that when the inflation factor is small, as in Figure 3.19, set at 5 for both male (M) and female (F) sexes, most of the contaminated samples are clustered closer to the cloud of

non-contaminated samples. However, a few contaminated samples are observed farther away from this cloud.

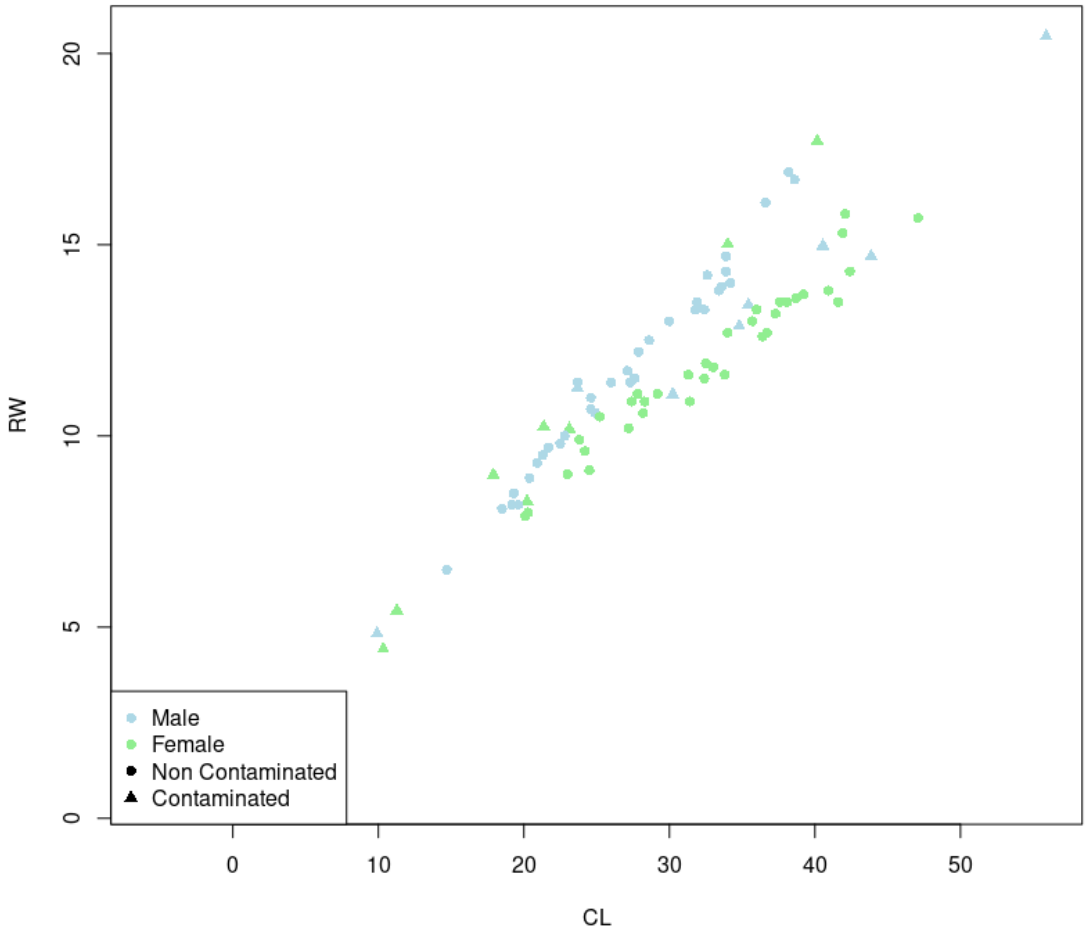



Figure 3.19: Pairs plot for carapace length CL and rear width RW with added contaminated samples in the training set using $\alpha_M = \alpha_F = 0.75, \eta_M = \eta_F = 5$

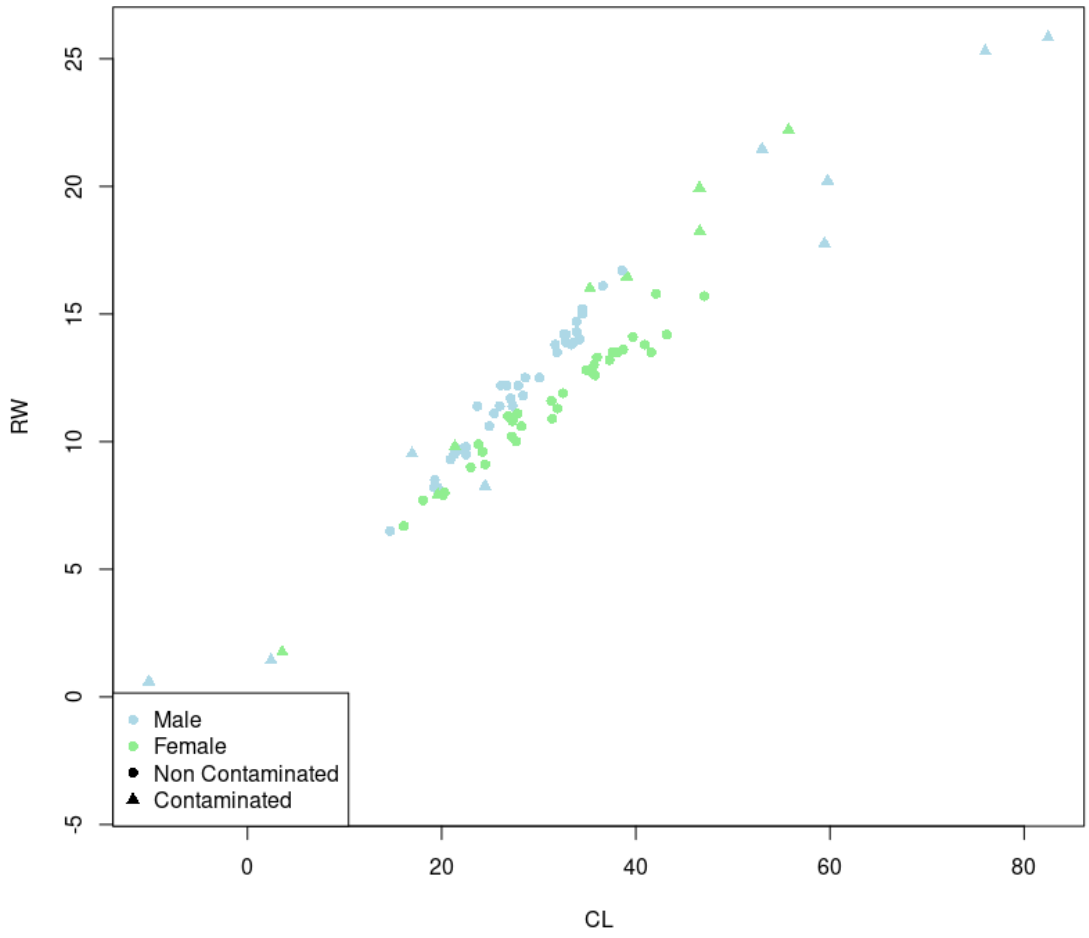



Figure 3.20: Pairs plot for carapace length CL and rear width RW with added contaminated samples in the training set using $\alpha_M = \alpha_F = 0.75, \eta_M = \eta_F = 10$

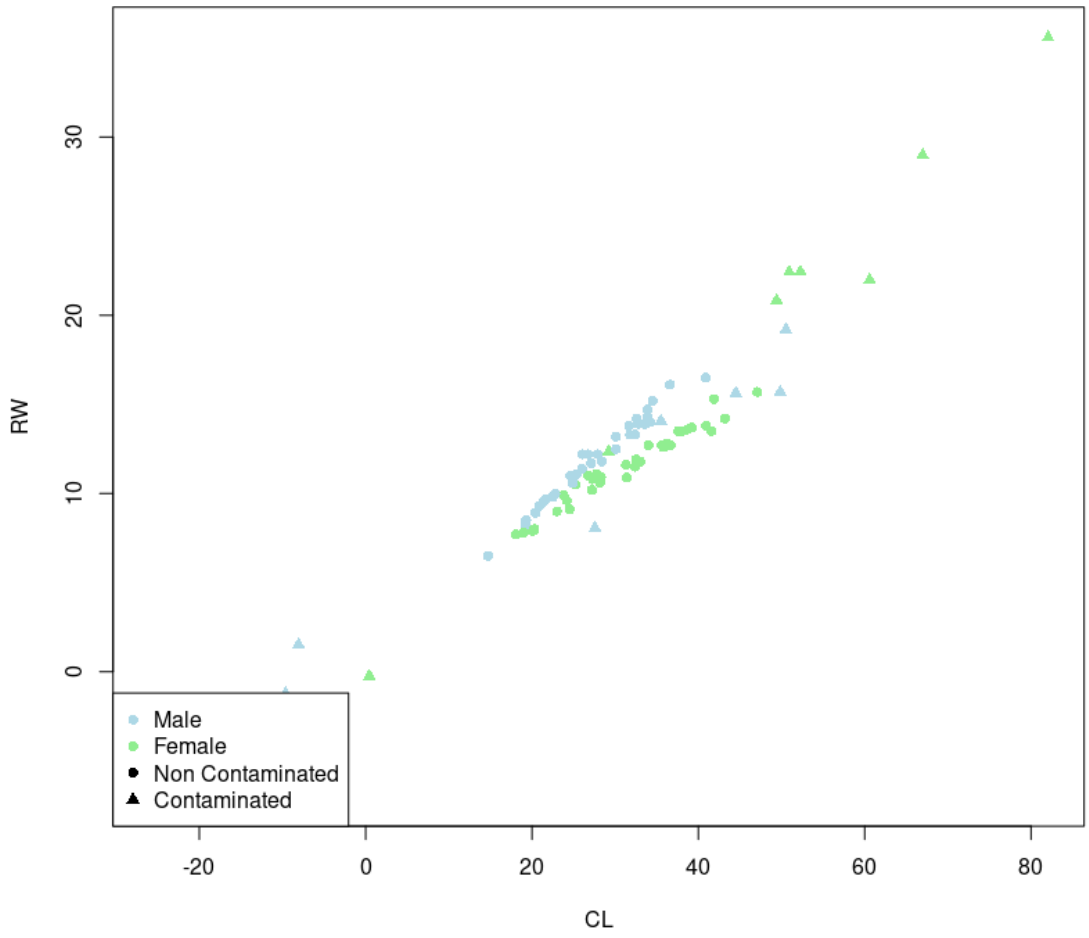



Figure 3.21: Pairs plot for carapace length CL and rear width RW with added contaminated samples in the training set using $\alpha_M = \alpha_F = 0.75, \eta_M = \eta_F = 15$

The contaminated samples move further away from the clusters of non-contaminated samples as $\eta$ increases (Figures 3.19, 3.20, 3.21). In addition, as the inflation factor increases from 5 to 10 and 15, some contaminated observations from males are mapped close to the cluster of non-contaminated females, and vice versa, posing an additional challenge to identify class membership quite apart from contamination status. Although the groups remain mostly separable, they become closer together.

It is expected that the model will be able to classify those contaminated observations that are far from the cluster of non-contaminated observations. However, identifying contaminated samples that are close to the cluster of non-contaminated samples poses a

challenge, and the model may not be expected to accurately identify them.

The blue crab samples are divided into 70% for training and 30% for testing models. This split will be utilised for the other datasets in this chapter as well. Since this is done randomly, the proportion of non-contaminated samples is approximately the same in both the training and test sets.

Differences in correct classification rate (CCR) and sensitivity are assessed to determine the accuracy of the models in classifying samples and contaminated samples in the test set. Positive differences indicate that the model using selected variables performs better than a model using all variables, whereas negative differences suggest that the model using all variables performs better than one using selected variables.

There were simulations in which the selected model failed to identify any of the contaminated samples, resulting in sensitivity values being indeterminate, as both true positives (TP) and false negatives (FN) were zero (see Equation 2.3). In such cases, sensitivity is assigned a value of zero since the selected model was unable to identify any contaminated samples. As mentioned previously, it appears that the variables $CL$ and $RW$ serve to differentiate between the sexes. So it is of interest to see how often they are selected by the variable selection.

To illustrate the effect of contaminated samples in the scenario introduced in Figure 3.20, the parameter estimates with and without the contaminated samples are computed. Also, a linear discriminant analysis is conducted to predict contaminated samples and ignoring class labels.

In Table 3.13 the parameter estimates assuming two groups with different mean but same variance are shown. It is visible that group means and most of the elements of the covariance matrix are slightly higher in the presence of contamination compared with the estimates in a non-contaminated scenario. Next, a linear discriminant analysis was fit to the contaminated dataset to obtain a discrimination rule based on the features to identify contaminated samples.

Table 3.13: Parameter estimates in the absence and presence of contaminated samples in crab data with parameters $\alpha_M = \alpha_F = 0.8$ and $\eta_M = \eta_F = 5$

**Estimates**

| Dataset | $\mu_M$ | $\mu_F$ | $\Sigma$ |
|---|---|---|---|
| without contamination | $\begin{pmatrix} 14.67 \\ 11.53 \\ 31.63 \\ 36.37 \\ 13.14 \end{pmatrix}$ | $\begin{pmatrix} 13.28 \\ 12.13 \\ 28.04 \\ 32.64 \\ 11.75 \end{pmatrix}$ | $\begin{pmatrix} 9.23 & 6.28 & 21.04 & 23.94 & 9.19 \\ 6.28 & 5.26 & 14.28 & 16.41 & 6.31 \\ 21.04 & 14.28 & 48.44 & 55.00 & 21.10 \\ 23.94 & 16.41 & 55.00 & 62.64 & 24.01 \\ 9.19 & 6.31 & 21.10 & 24.01 & 9.33 \end{pmatrix}$ |
| with contamination | $\begin{pmatrix} 15.52 \\ 12.30 \\ 33.60 \\ 38.69 \\ 14.05 \end{pmatrix}$ | $\begin{pmatrix} 13.64 \\ 12.42 \\ 29.09 \\ 33.69 \\ 12.34 \end{pmatrix}$ | $\begin{pmatrix} 20.08 & 15.10 & 47.37 & 54.25 & 22.03 \\ 15.10 & 12.64 & 34.74 & 40.04 & 16.57 \\ 47.37 & 34.74 & 109.42 & 124.88 & 51.12 \\ 54.25 & 40.04 & 124.88 & 143.13 & 58.45 \\ 22.03 & 16.57 & 51.12 & 58.45 & 24.63 \end{pmatrix}$ |

The confusion matrix obtained by a linear discriminant analysis model that aims to identify contaminated samples is shown in Table 3.14. The model struggles to identify contaminated samples in the train set since it only detected 7 out of 24 and failed in identifying 17 contaminated samples in the test set. In the test set, the linear discriminant analysis model cannot identify any of the contaminated samples. This can be seen clearly looking at the linear discriminant analysis function in Figure 3.22, since there is an overlapping between the the contaminated and non-contaminated observations.

Table 3.14: Confusion matrices of a LDA model for identifying contaminated samples

(a) Train set

| | | **Predicted** | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Actual** | **Negative** | 70 | 0 |
| | **Positive** | 17 | 7 |

(b) Test set

| | | **Predicted** | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Actual** | **Negative** | 30 | 0 |
| | **Positive** | 10 | 0 |

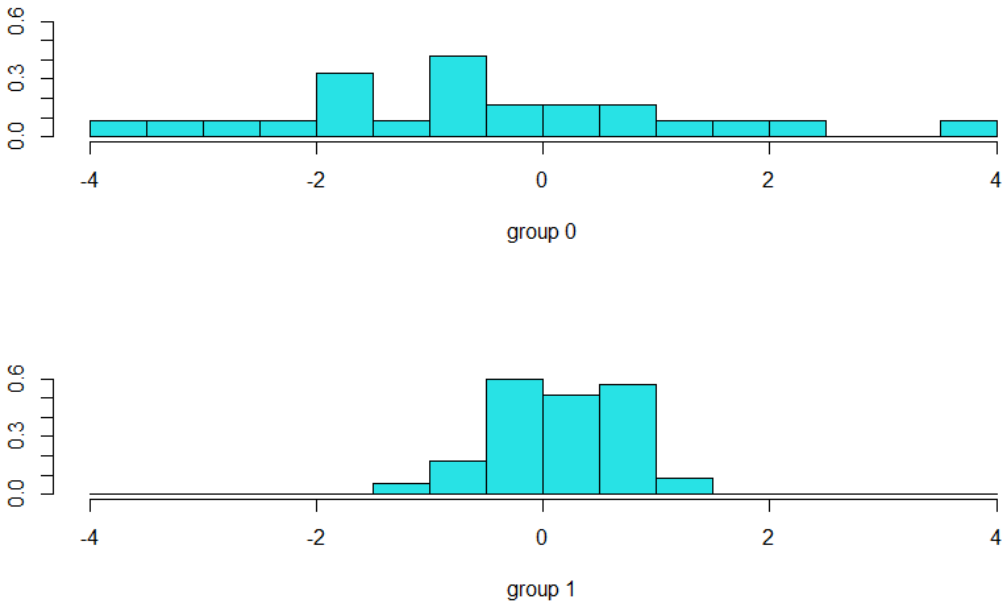



Figure 3.22: Histogram for discriminant function values

Fitting a variable selection supervised mixture of contaminated Gaussian distributions to the contaminated training data displayed in Figure 3.20 produced the subset of variables "RD", "RW", and "FL" with a confusion matrix shown in Table 3.15. From the confusion matrix is clear that the variable selection model combined with a mixture of contaminated Gaussian distribution yields higher level of contamination sensitivity in the training and test sets since it is able to identify more contaminated samples than the linear discriminant analysis model.

Table 3.15: Confusion matrices of a mixture of contaminated Gaussians for identifying contaminated samples

(a) Train set

| | | **Predicted** | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Actual** | **Negative** | 70 | 0 |
| | **Positive** | 13 | 11 |

(b) Test set

| | | **Predicted** | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Actual** | **Negative** | 30 | 0 |
| | **Positive** | 4 | 6 |

Table 3.16 shows a comparison between the variable selection for a supervised contam-

inated mixture of Gaussian distributions and a linear discriminant analysis model (LDA) in CCR, sensitivity, and specificity. The results show a superiority of the variable selection (Var. Sel.) model over the linear discriminant analysis model. The LDA model displays lower contamination correct classification rate for the training and test sets (0.82 and 0.75, respectively) compared to the model with variable selection (0.86 and 0.90, respectively). A similar pattern is observed for the LDA model that displays lower contamination sensitivity in the training and test sets (0.29 and 0, respectively) compared to the model with variable selection (0.46 and 0.60, respectively). In terms of contamination specificity there were no differences between both models in the training and test sets. In this particular scenario, the model with variable selection yields better performance in contamination prediction. This is due to the effect that the contaminated samples have on the parameter estimates of the linear discriminant analysis model. Additionally, the data required a more complex model than the linear discriminant model that assumes same variance covariance matrix for all classes, and the better model that fits the data from the models (EII, VII, EEI, VEI, EEE, VVV) was a VVV.

Table 3.16: Comparison between a linear discriminant analysis model and variable selection for a mixture of contaminated Gaussian distributions in correct classification rate. sensitivity, and specificity for crab data

| Metrics | Train | | Test | |
|---|---|---|---|---|
| | LDA | Var. Sel. | LDA | Var. Sel. |
| CCR | 0.82 | 0.86 | 0.75 | 0.90 |
| Sensitivity | 0.29 | 0.46 | 0 | 0.60 |
| Specificity | 1 | 1 | 1 | 1 |

In the following sections in this chapter, the results for the class correct classification rate and test contamination sensitivity are displayed. The former plays an important role in selecting a variable for inclusion while the latter is the property of the mixture of contaminated Gaussian that is desired to extend to high-dimensional data.

### 3.5.2 Results

Initially, an analysis is conducted to inspect the variables selected by the greedy algorithm, calculating both the average number of variables and the frequency of each variable's

inclusion in the selected models. The results in Table 3.17a suggest that there is some variability in variable selection. 45% of the models include at most three of the five variables. However, the majority of selected models consist of four (32%) or three (28%) variables. Thus, in 23% of the simulations, all the variables are selected, while in the remaining 77%, a smaller model is preferred. The variables RW and CL are the most frequently included in the chosen model, appearing in 808 (99.75%) and 749 (92.47%) of the 810 simulations, respectively as shown in Table 3.17b.

Table 3.17: Frequency and percentage of variables' selection and number of variables selected across all crab plasmode datasets

(a) Frequency and % of number of variables selected across all crabs plasmode datasets

| No. of selected variables | Frequency | % |
|---|---|---|
| 2 | 136 | 17% |
| 3 | 230 | 28% |
| 4 | 257 | 32% |
| 5 | 187 | 23% |

(b) Frequency and % of variables' selection across all crabs plasmode datasets

| Variable | Frequency | % |
|---|---|---|
| RW | 808 | 99.75% |
| CL | 749 | 92.47% |
| CW | 489 | 60.37% |
| FL | 472 | 58.27% |
| BD | 4007 | 50.25% |

Figure 3.23 gives a boxplot of the correct classification rate for the classification of test samples and the contamination sensitivity of the test sample. It illustrates a modest enhancement classifying new blue crab samples into male and female with selected variables compared to the use of all variables. The test class correct classification rate (CCR) is higher on average for models employing variable selection. However, a decrease in contamination sensitivity is also observed compared to a full model, which includes all the variables.

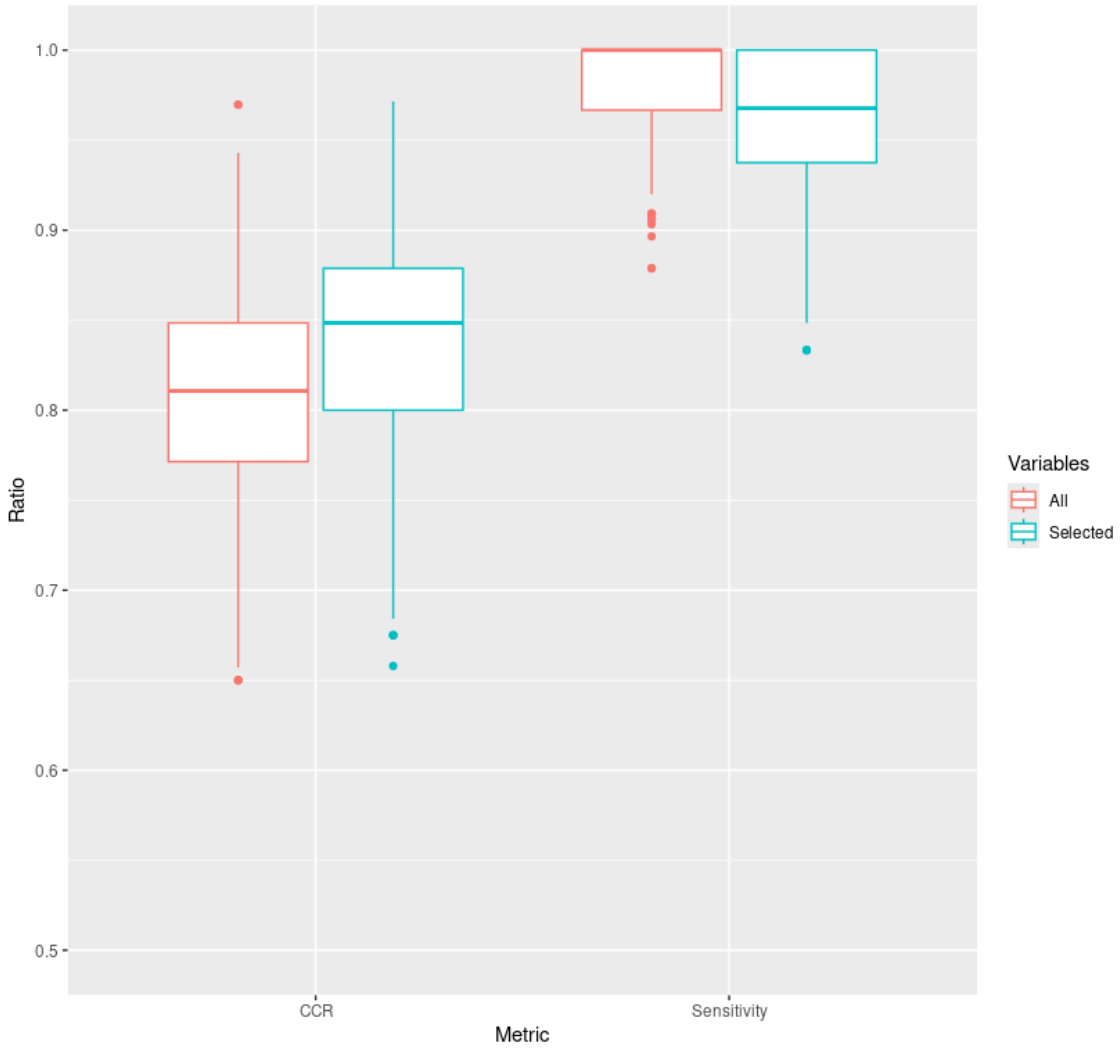



Figure 3.23: Class correct classification rate and contamination sensitivity by variable subset in the test crabs dataset

In Figure 3.24 we look at the differences in class correct classification rate for selected variables versus all variables split according to whether the simulations have classes with the same proportion of contamination or differing. Whether $\alpha$ is the same across classes or varies, there is mostly a positive but not a big difference between employing a subset of selected variables or utilising the entire set of variables for test class correct classification rates. There is a marginal improvement in test class correct classification rate but a slight decrease in test contamination sensitivity, particularly when the percentage of non-contaminated samples $\alpha$ remains equal across the classes. Sensitivity tends to deteriorate slightly more when using selected variables irrespective of $\alpha$'s being the same or different across classes.

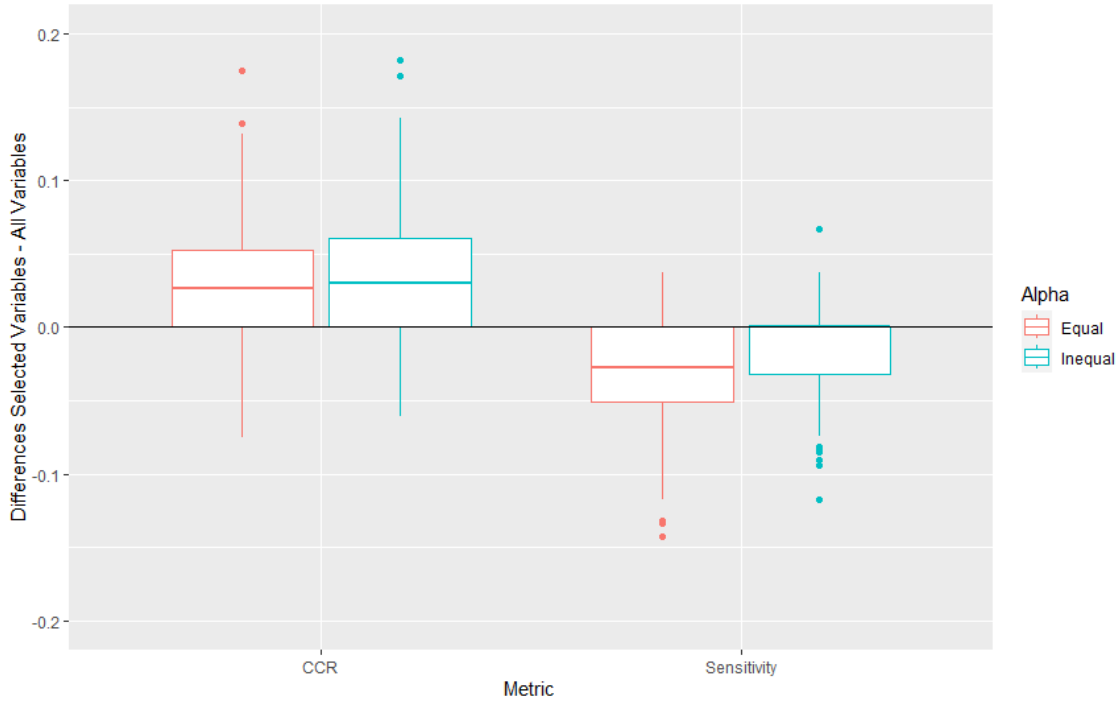



Figure 3.24: Difference in class correct classification rate and contamination sensitivity between models using selected and all variables for identifying contaminated blue crabs across various values of $\alpha$ for test crab data

In Figure 3.25 we have the difference in CCR and Sensitivity between selected variables versus all variables split according to whether the simulations have classes with the same magnitude of contamination or differing. Analysis of $\eta$ yields similar results to $\alpha$. Using the selected model enhances the accuracy of classifying the sex of a new blue crab sample compared to the all-variable model. However, there is a decrease in the ability to identify contaminated samples, as sensitivity indicates negative differences, regardless of whether $\eta$ is consistent across males and females.

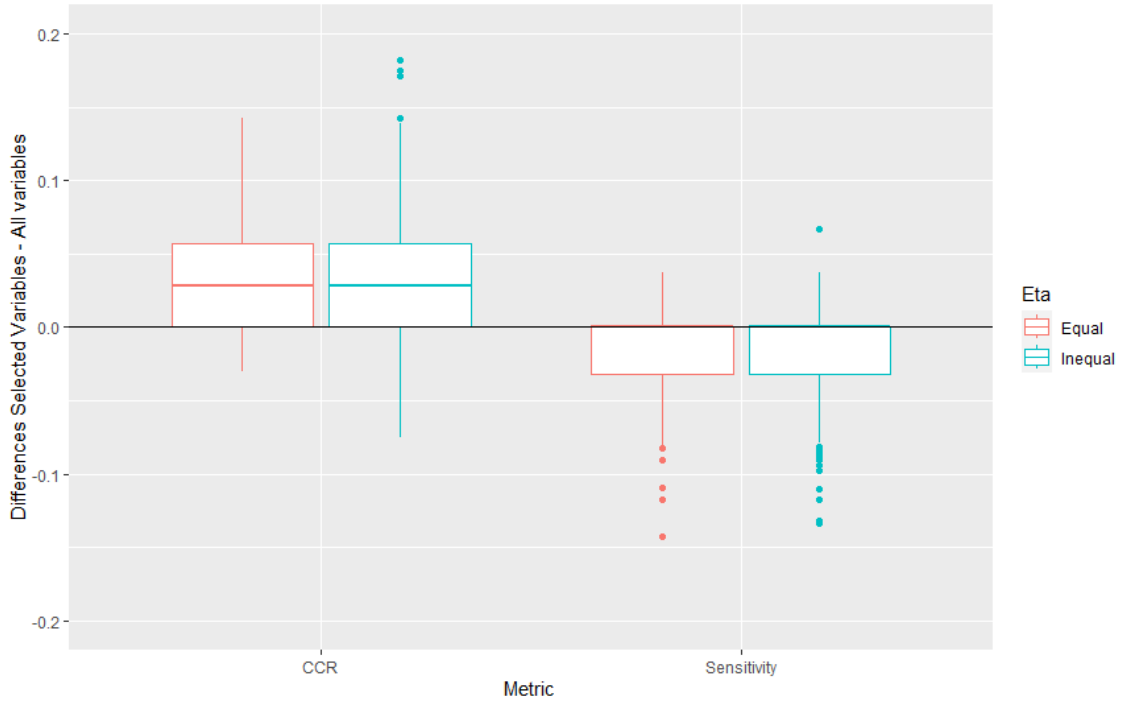



Figure 3.25: Difference in class correct classification rate and contamination sensitivity between models using selected and all variables for identifying contaminated blue crabs across various values of $\eta$ for test crab data

In conclusion, in datasets with highly correlated variables, such as the crab dataset, the proposed method slightly harms sensitivity of identifying contaminated observations when contamination occurs across all variables (both selected and non-selected). It is anticipated that, due to the redundancy suggested by high correlations, a smaller subset of variables will be enough to classify new samples. Hence, it's unsurprising to observe that models utilising the selected variables outperform the model with all variables in terms of classification.

## 3.6 Wine data

The wine dataset Forina et al. (1988) is made up measurements of 13 physical and chemical properties of 178 bottles of wines originating from the same Italian region. These bottles represent three distinct cultivars: Barbela (48 bottles), Barolo (59 bottles), and Grignolino (71 bottles). Initially published by Forina et al. (1988), the dataset has since been analyzed by Aeberhard and Forina (1991), Spadaro et al. (2010), Hurley (2004),

Von Weinen (1986) and Wehrens (2011). Azzalini (2013) noted a slight indication of asymmetric distribution in the *Phenols* content of Barolo. Bouveyron C (2019) explained the impracticality of fitting a model without restrictions on the variance-covariance matrices due to overparametrization, suggesting a covariance structure model EII. Moreover, this dataset exhibits an imbalance in the proportion of samples corresponding to each wine type.

The objective is to observed the performance of the proposed method in comparison to its alternative without variable selection particularly in a scenario involving imbalanced classes, correlated variables, and an asymmetrical variable.

Figure 3.26 illustrates a strong positive correlation between the variables $Flavanoids$ and $Phenols$, $Dilution$ and $Phenols$, as well as $Flavanoids$. Additionally, $Phenols$ and $Flavanoids$ exhibit medium positive correlations with other variables, such as $Hue$, $Proline$, $Nonflavanoid$, and $Proanthocyanins$. Negative correlations between $Malic$ and $Hue$, $Color$ and $Hue$, $Nonflavanoid$ with $Dilution$, $Phenols$, and $Flavanoids$. From Figure 3.27 it is evident that potential variables for distinguishing between wine types include $Color$, $Hue$, and $Flavanovids$.

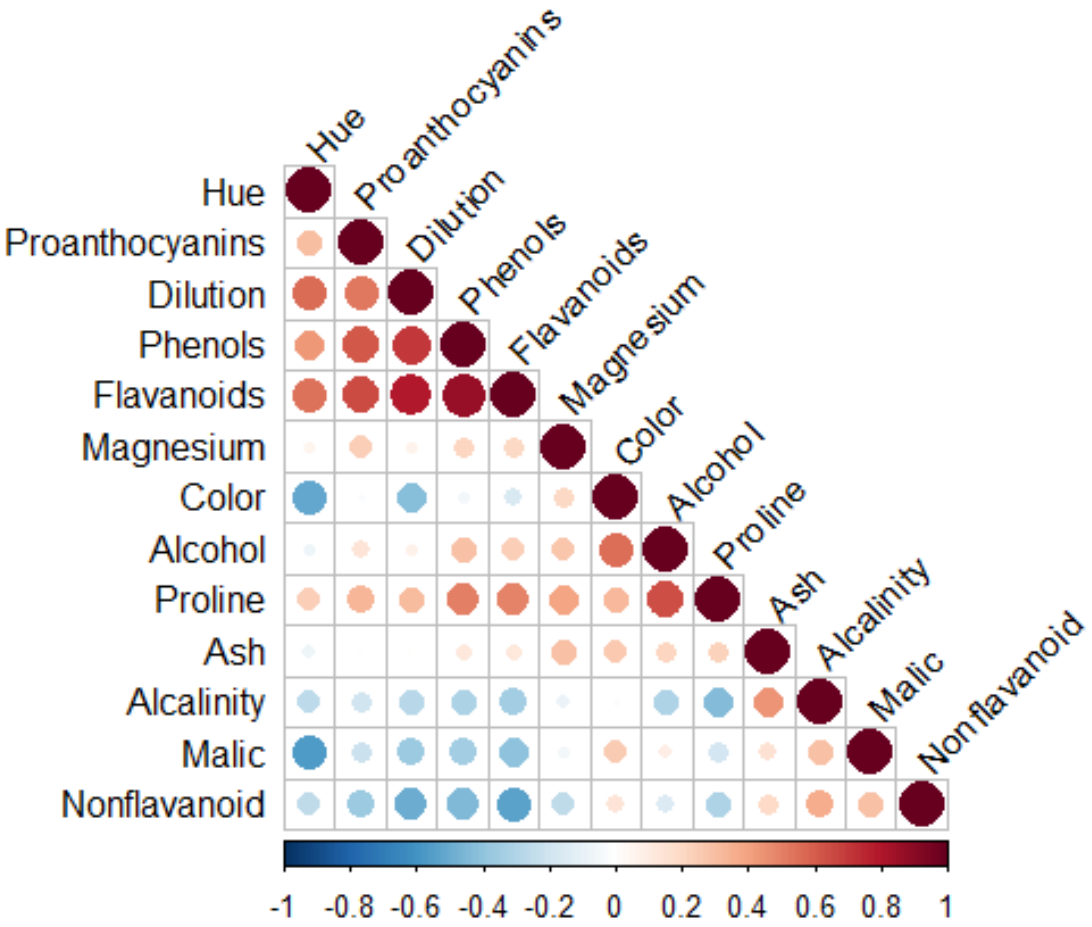



Figure 3.26: Correlation matrix of non-contaminated wine data

In Figure 3.27, looking at the class dispersion for the variables that offer a better separation, it is noticed that the classes are quite close. Although a separation is possible, for example using the *Color* and *Alcohol* variables, there is considerable overlap in at least two of the three types of wines at least in 2 dimensions.

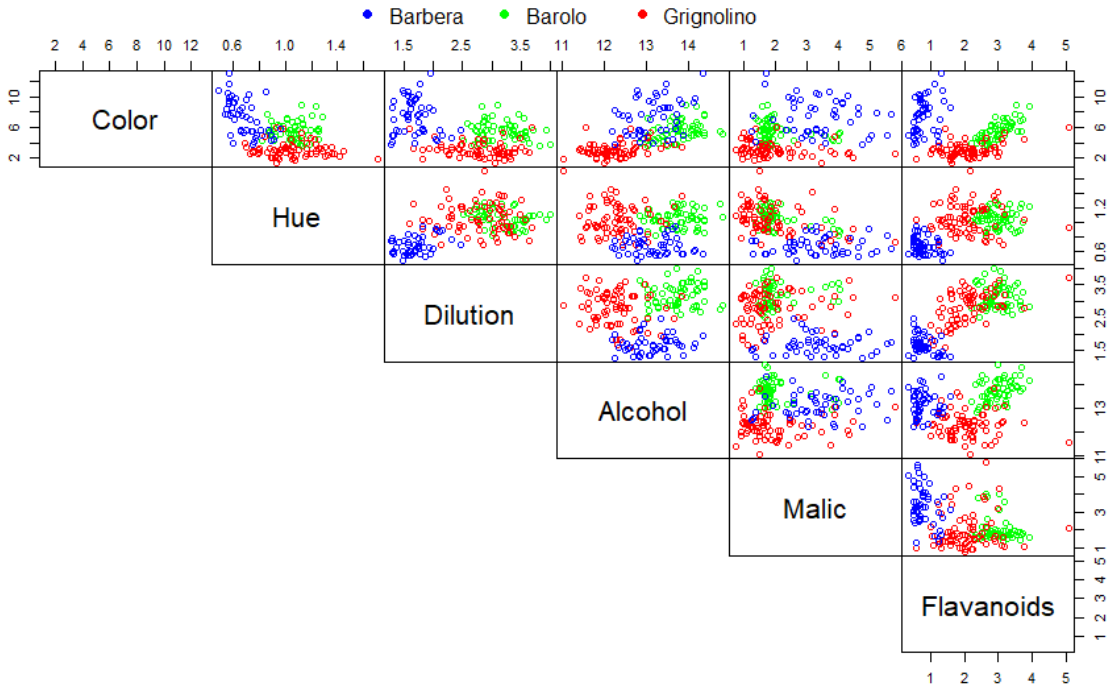



Figure 3.27: Pairs plot of original wine data with color denoting region of origin: wine bottles from Barbera region are blue, Barolo region green & Grignolino red.

### 3.6.1 Contaminating only variable *color* in wine data

To make the problem more difficult in the wine dataset, the contaminated observations were simulated in such a way that the contamination/variable inflation factor only affected only one of the variables, color. This was one of the variables that creates a separation between classes with the aim of assessing the performance of the proposed method in such conditions.

In the wine data, there are three types of wine. The contamination of each type of wine $g$ is controlled by two parameters: the percentage of non-contaminated samples $F7$ ($\alpha_g$) and the inflation factor of the variance $F8$ ($\eta_g$). These parameters are assumed to be independent and each of them is allowed to take three values (see Table 3.18). The possible combinations of parameter values in each type lead to 729 scenarios and, since each of these scenarios is simulated 10 times, the overall number of simulations is 7290.

In real data sets, having a small proportion of contaminated samples is common. Consequently, it is realistic to consider higher values for the proportion of non-contaminated samples ($\alpha$) as it is shown in Table 3.18 where the levels could be labelled as a high (75%), medium (80%), and low degree of contamination (85%). The variable inflation factor $\eta$

was modelled to affect only the variable *Color*.

Table 3.18: Simulated levels for factors percentage of non-contaminated samples $\alpha$ and variance inflation factor $\eta$

| Factors | Description | Levels |
|---|---|---|
| $F_7$ | Percentage of non-contaminated samples | (75%,80%,85%). |
| $F_8$ | Variance inflation factor | (5,10,15). |

As the inflation factor F8 increases, contaminated samples move away from the cloud of non-contaminated samples (See Figures 3.19-3.21). Then, it is interesting varying $\eta$ from small ($\eta = 5$), medium ($\eta = 10$), and high dispersion ($\eta = 15$) of contaminated samples.

In Figures 3.28, 3.29, 3.30 we have pairsplots of plasmode wine data with 20% induced contamination and amount of contamination in the variable color ranging over 5, 10 or 15. There are pairs of variables that can provide a separation of classes ignoring contaminated samples with a certain level of overlap, such as *Flavanoids* and *Dilution* or *Malic* and *Dilution* to mention a few. This separation is less clear when contaminated samples are taken into account and the inflation factor is small (see Figure 3.28). However, increasing the inflation factor causes contaminated samples to move away from the cloud of points corresponding to their class, but they may appear close to the cloud of points corresponding to another class, which makes it difficult to achieve the two goals that are correct class classification and identification of contamination in new samples using only a pair of variables.

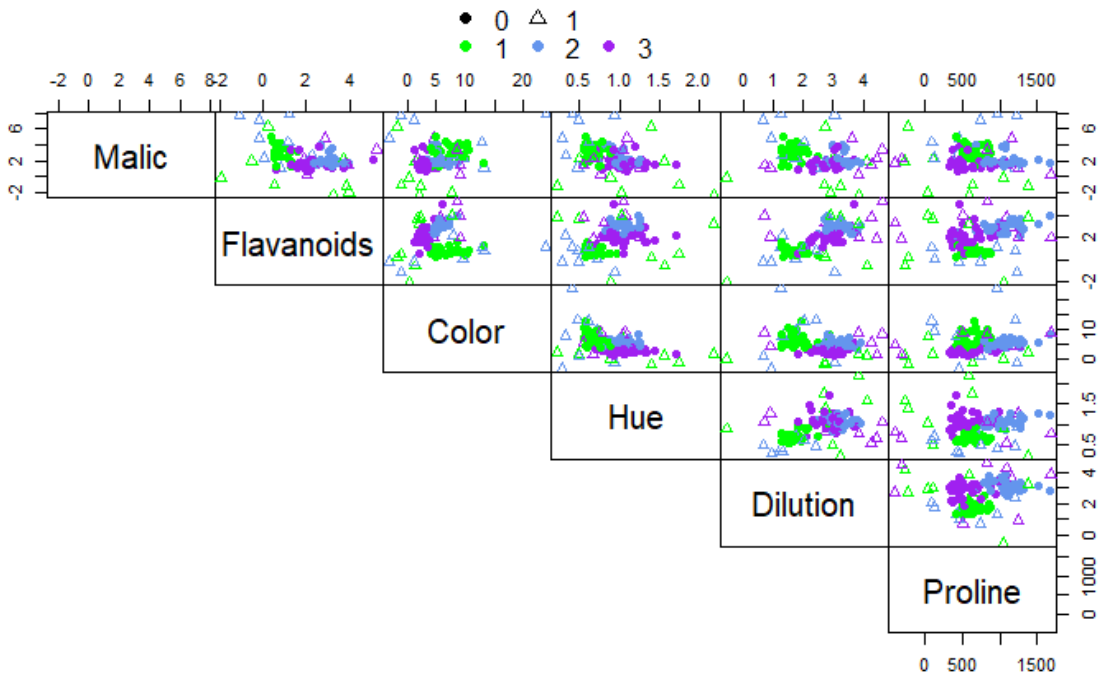



Figure 3.28: Pairs plot of plasmode wine data with Barbera in blue, Barolo region in green & Grignolino in red; (•) denoting uncontaminated & triangles (△) representing contaminated specimens where contaminated settings were: $\alpha = 80\%$ and $\eta = 5$ (for variable color) for all types of wine

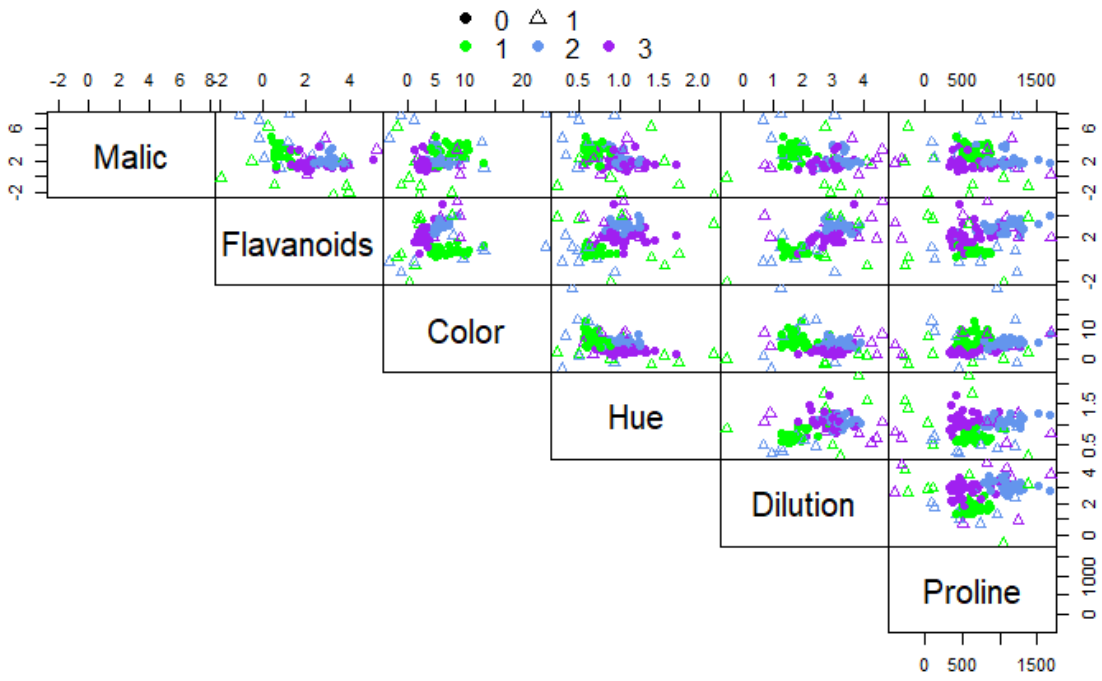



Figure 3.29: Pairs plot of plasmode wine data with Barbera in blue, Barolo region in green & Grignolino in red; (•) denoting uncontaminated & triangles (△) representing contaminated specimens where contaminated settings were: $\alpha = 80\%$ and $\eta = 10$ (for variable color) for all types of wine

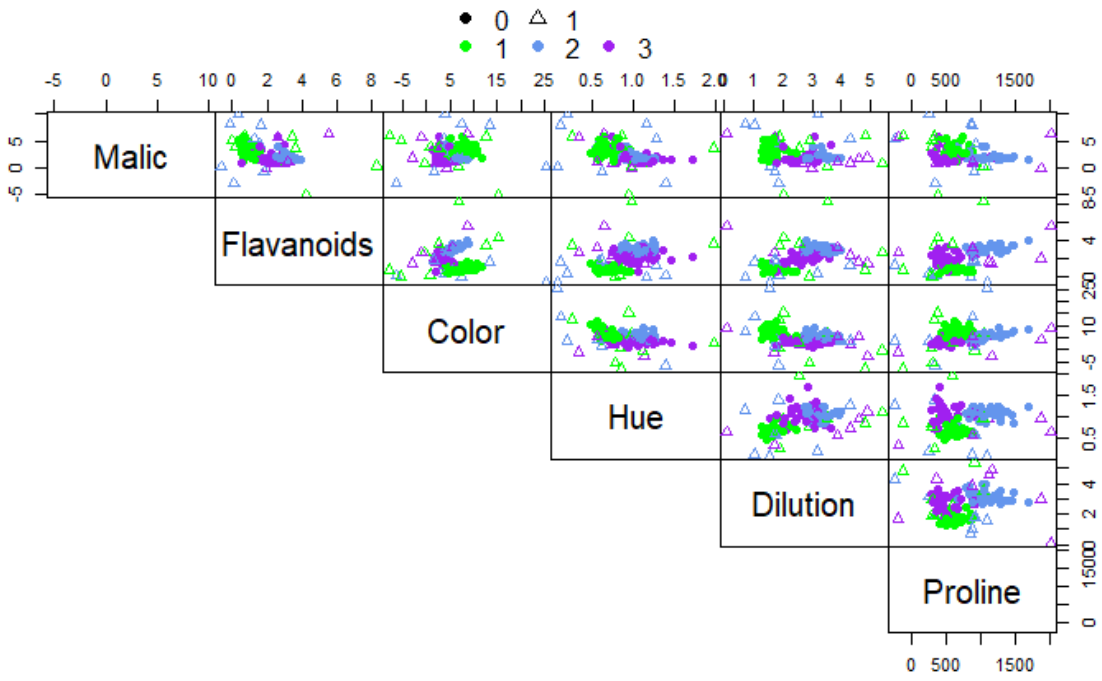



Figure 3.30: Pairs plot of plasmode wine data with Barbera in blue, Barolo region in green & Grignolino in red; (•) denoting uncontaminated & triangles (△) representing contaminated specimens where contaminated settings were: $\alpha = 80\%$ and $\eta = 15$ (for variable color) for all types of wine

### 3.6.2 Results

**Results of contaminating all variables and contaminating only variable color in wine data**

Looking at the variables more frequently selected by the variable selection algorithm in Table 3.19b, it is clearly seen that *Color*, *Flavanoids*, *Alcohol*, *Hue*, and *Dilution* are at the top. For example, *Color* is chosen in 2598 in the selected model 2598(98%) and *Flavanoids* 2553(96%) of the total of 2790 simulations . The most frequent numbers of the variables selected in Table 3.19a were $4, 5, 6$ variables in $22\%, 27\%, 19\%$ of the total plasmode datasets.

Table 3.19: Frequency and percentage of variables' selection and number of variables selected across all wine plasmode datasets

(a) Frequency and % of number of variables selected for all wine plasmode datasets

| No. of selected variables | Frequency | % |
|---|---|---|
| 2 | 79 | 3% |
| 3 | 356 | 13% |
| 4 | 584 | 22% |
| 5 | 708 | 27% |
| 6 | 515 | 19% |
| 7 | 248 | 9% |
| 8 | 113 | 4% |
| 9 | 37 | 1% |
| 10 | 8 | 0% |
| 11 | 2 | 0% |

(b) Frequency and % of variables' selection for all wine plasmode datasets

| Variable | Frequency | % |
|---|---|---|
| Color | 2598 | 98% |
| Flavanoids | 2553 | 96% |
| Alcohol | 1940 | 73% |
| Hue | 1744 | 66% |
| Dilution | 1286 | 49% |
| Proline | 848 | 32% |
| Ash | 517 | 20% |
| Malic | 452 | 17% |
| Phenols | 419 | 16% |
| Nonflavanoid | 275 | 10% |
| Proanthocyanins | 260 | 10% |
| Magnesium | 221 | 8% |
| Alcalinity | 154 | 6% |

It is apparent in Figure 3.31, that employing variable selection alongside a mixture of contaminated Gaussians yields comparable correct classification rates to models incorporating all variables when looking at all possible plasmode datasets. However, including all variables surpasses using only selected variables in sensitivity for contamination detection. This is expected, since the color variable, where the contamination lies, is only guaranteed to be included when using all variables.

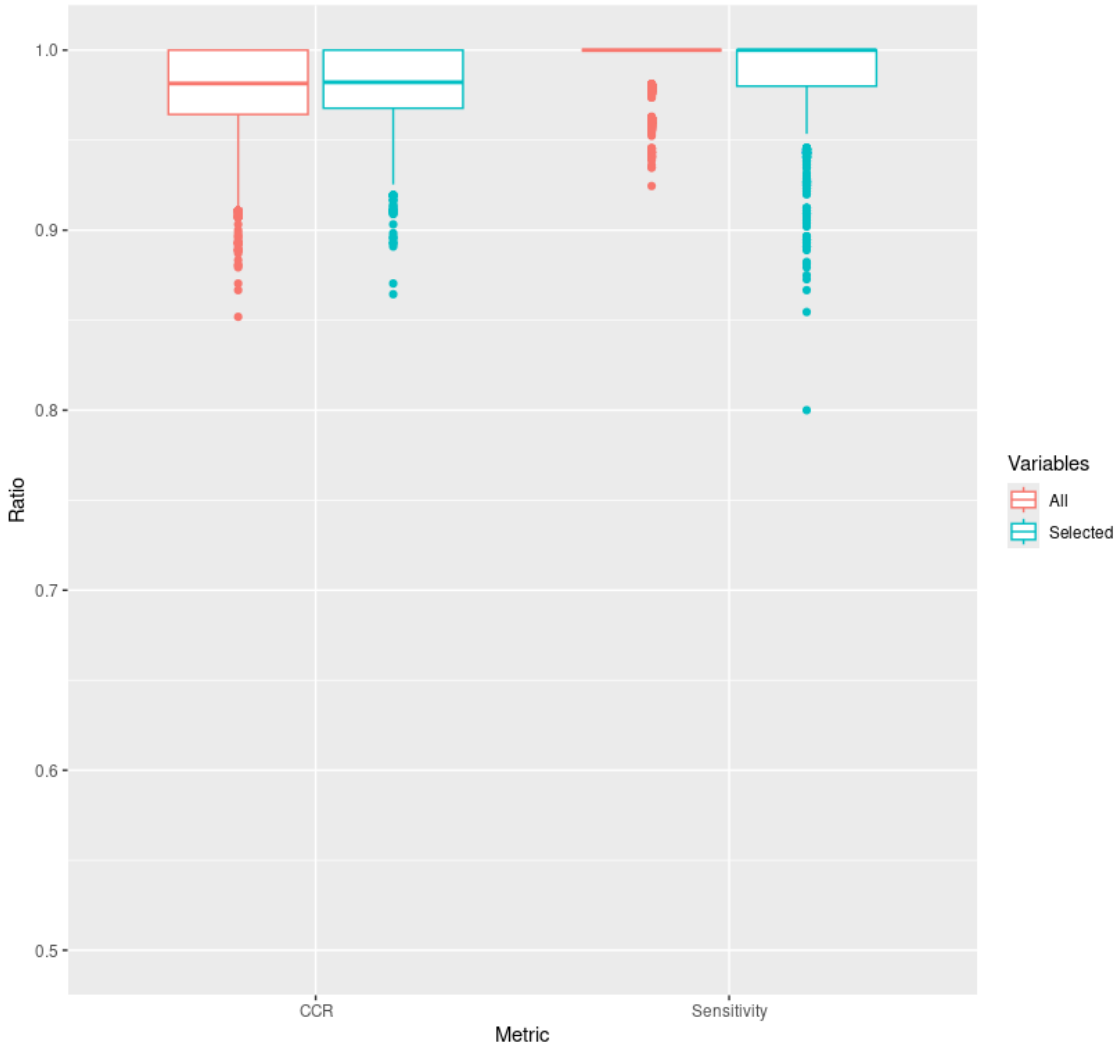



Figure 3.31: Class correct classification rate and contamination sensitivity by variable subset in the test wine dataset

In the wine dataset that was contaminated, it is observed that there is a marginal improvement using the selected variables instead of all the variables in terms of CCR when $\alpha$ was different across classes. In terms of test contamination sensitivity it is visible that the model composed of the selected variables seems to obtain a slightly lower performance in comparison with a model that incorporates all the variables regardless of the chosen level of $\alpha$ (see Figure 3.31).

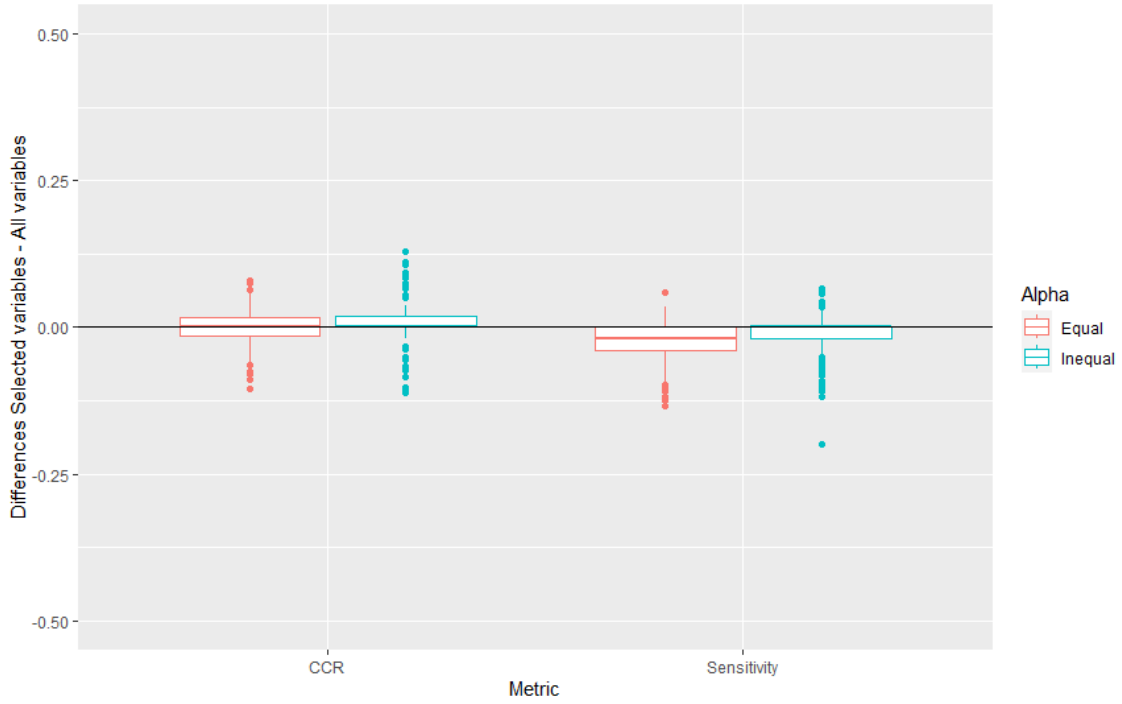



Figure 3.32: Difference in class correct classification rate and contamination sensitivity between models using selected and all variables across various values of $\alpha$ for test wine data

In the wine dataset affected by contamination, a marginal improvement is evident when employing selected variables instead of incorporating all variables, as observed in terms of the class Correct Classification Rate (CCR), regardless of whether $\eta$ is the same or different across classes. In terms of contamination sensitivity, models comprising selected variables exhibit slightly lower performance compared to those incorporating all variables, regardless of the chosen level of $\eta$ (refer to Figure 3.33).

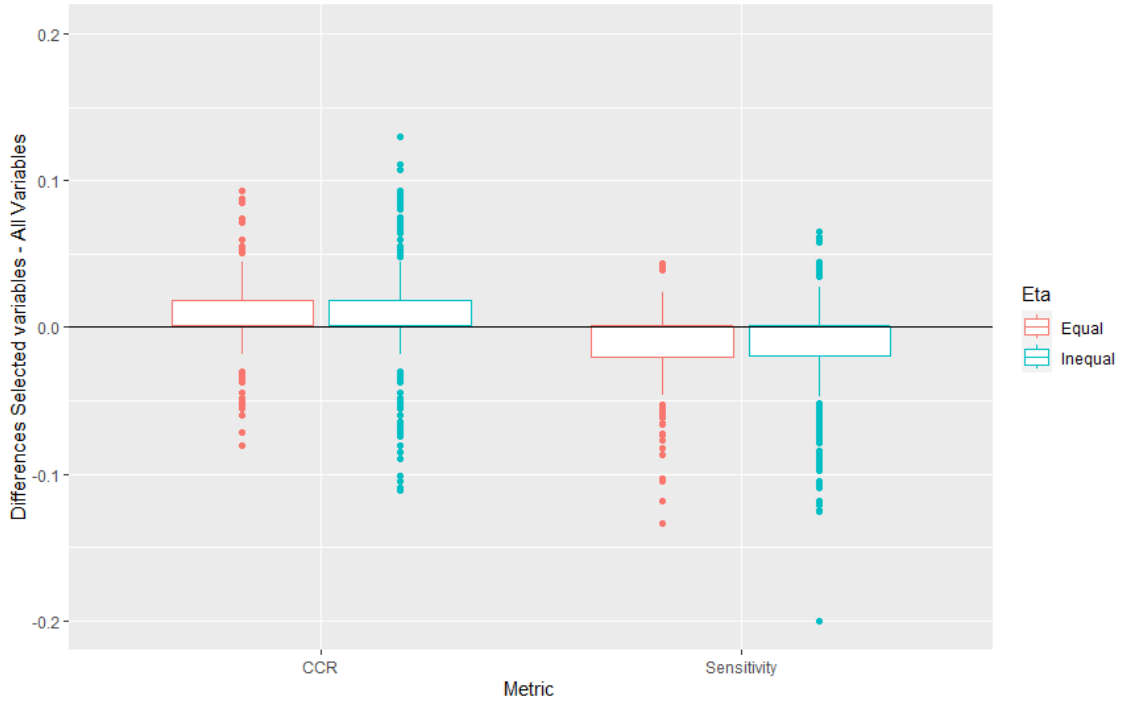



Figure 3.33: Difference in class correct classification rate and contamination sensitivity between models using selected and all variables across various values of $\eta$ for test wine data

Summarising the results obtained by contaminating only the feature *Color*, it is evident that similar outcomes are observed. There was an anticipation that the contamination of the variable *Color* would significantly impact the selected models, given its frequent selection. However, the contamination of *Color* has a marginal positive effect on the Correct Classification Rate (CCR) for both equal and unequal values of $\alpha$ and $\eta$. Conversely, the effect on test contamination sensitivity is predominantly negative. This phenomenon could be attributed to the exclusion of the variable *Color* in the subset of selected variables, thus rendering part of the contamination information it carries irreplaceable when not included in the selected model. Furthermore, if the variable *Color* is contaminated, metrics such as CCR and Sensitivity would be adversely affected, consequently reducing the overall mean, as can be observed in Table 3.20.

## 3.7 Diagnostic Wisconsin breast cancer data

The database contains data for 569 patients who were observed across 30 variables related to cell nuclei obtained from digitized images of a fine needle aspirate (FNA) of a

Table 3.20: Mean values of CCR, sensitivity, and specificity for models using variable selection excluding and including the variable *color* in the final model

| Metrics | Models excluding variable *Color* | Models including variable *Color* |
|---|---|---|
| CCR | 0.98 | 0.98 |
| Sensitivity | 0.98 | 1.00 |
| Specificity | 0.64 | 0.56 |

breast mass. Each cancer patient received a diagnosis of either malignant or benign. The dataset is unbalanced, comprising 357(63%) benign and 212(37%) malignant tumors. The Wisconsin Breast Cancer (Diagnostic) dataset is available at the UCI Machine Learning Repository and has been analysed by researchers such as Dubey et al. (2016) and Kadhim and Kamil (2022).

Looking closely at the correlation plot there are some strong positive correlations (See Figure 3.34). For example, *area_mean* is positively correlated with the *radius_mean* and *perimeter_mean* of the tumor. In addition to this, $concavity_extreme$ and $Nconcaye\_extreme$ are correlated with *compactness_mean*, *concavity_mean*, and $Nconcaye\_mean$. In general, there is a positive correlation between radius, perimeter, and area.

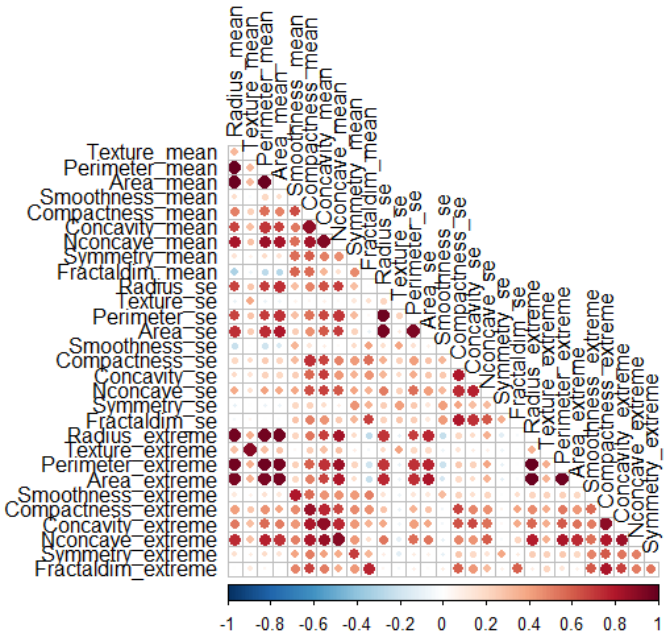



Figure 3.34: Correlation matrix of non-contaminated diagnostic Wisconsin breast cancer data

The variables that seem to offer better class separation between benign and malignant tumors in the breast after observing some of their pairs scatterplot are $Noncave\_extreme$, $Perimeter\_extreme$, $Perimeter\_mean$, $Radius\_extreme$, $Noncave\_mean$, $Area\_extreme$, $Concavity\_mean$, $Texture\_extreme$. Although separation of classes is possible, there is an overlapping of the two classes in a great majority of graphs of pairs of variables (see Figure 3.35).

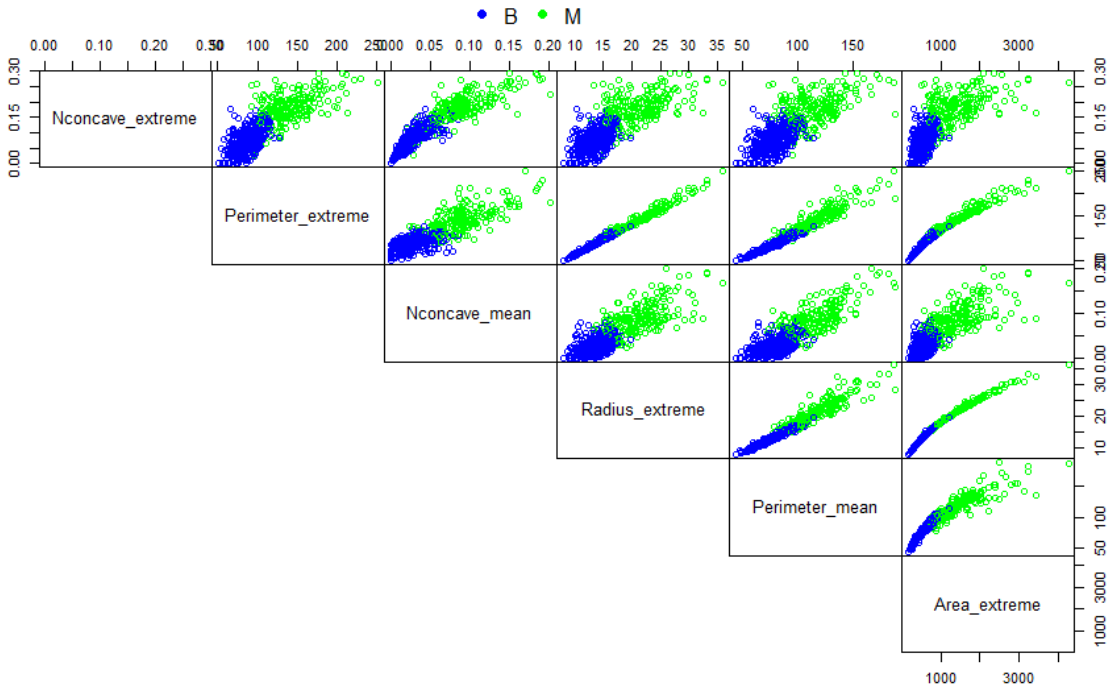



Figure 3.35: Uncontaminated Wisconsin breast cancer data variables

### 3.7.1 Contaminating diagnostic Wisconsin breast cancer data

Contaminated samples are added using the same procedure previously described for the crab data set and the variable inflation factor was the same within classes. In the diagnostic Wisconsin breast cancer data, there are two classes of tumors. The contamination of each type of tumor $g$ is controlled by the parameters introduced previously $F8(\alpha_g)$ and the inflation factor of the variance $F9(\eta_g)$. These parameters are independent and each of them can take three values (see Table 3.18). The possible combinations of parameter values in each type lead to 81 scenarios and since each of these scenarios is simulated 10 times. Therefore overall number of simulations is 810.

### 3.7.2 Results

#### Results diagnostic Wisconsin breast cancer data

In the variable selection process, it is expected to produce a model made up of a small number of variables. For this particular data set, the results suggest the size of the model goes from 7 to 10 54% of the time (see Figure 3.36). A desired property is variable selection is that the process produces very similar models for different instances of a contaminated

data set. It is evaluated by looking at the number of times that the variables were selected to be part of the classification model of the total number of simulations. Observing the variables most frequently selected to make up the classification model, it is found that *Nonconcave_extreme*, *Perimeter_extreme*, and *Perimeter_mean*, to mention a few, were selected 771(95%), 725(90%), and 607(75%) times in a total of 881 simulations.(see Figure 3.37).

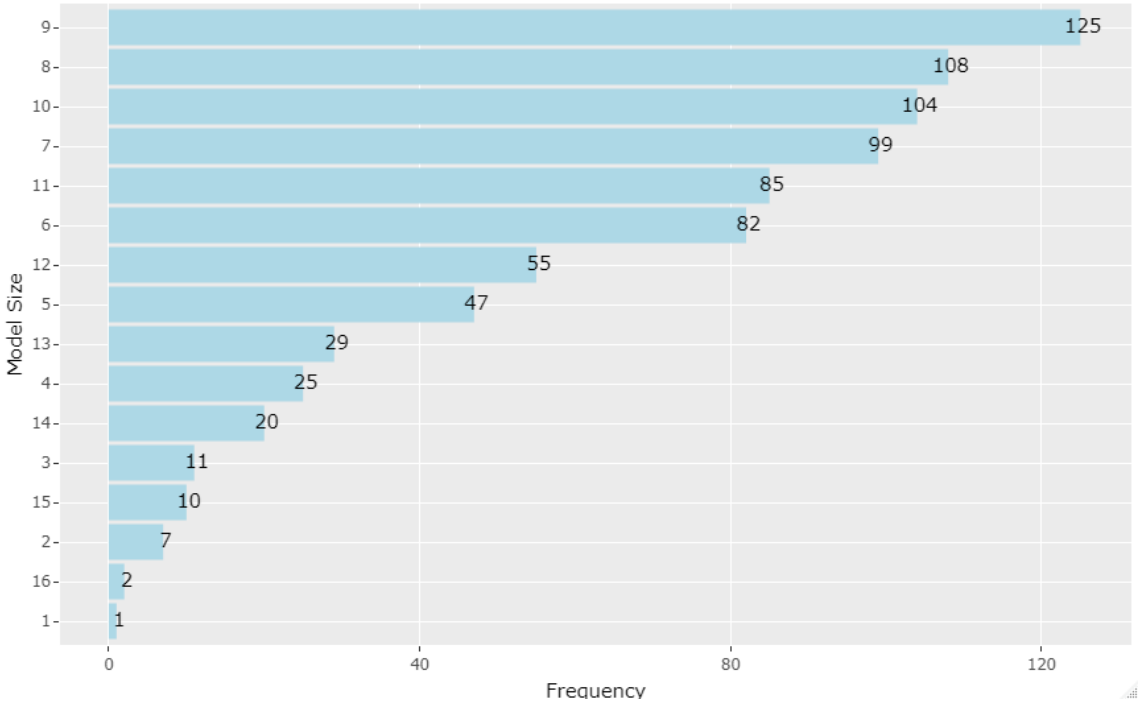



Figure 3.36: Frequency of number of variables selected by the greedy search algorithm forcontaminated Wisconsin breast cancer data

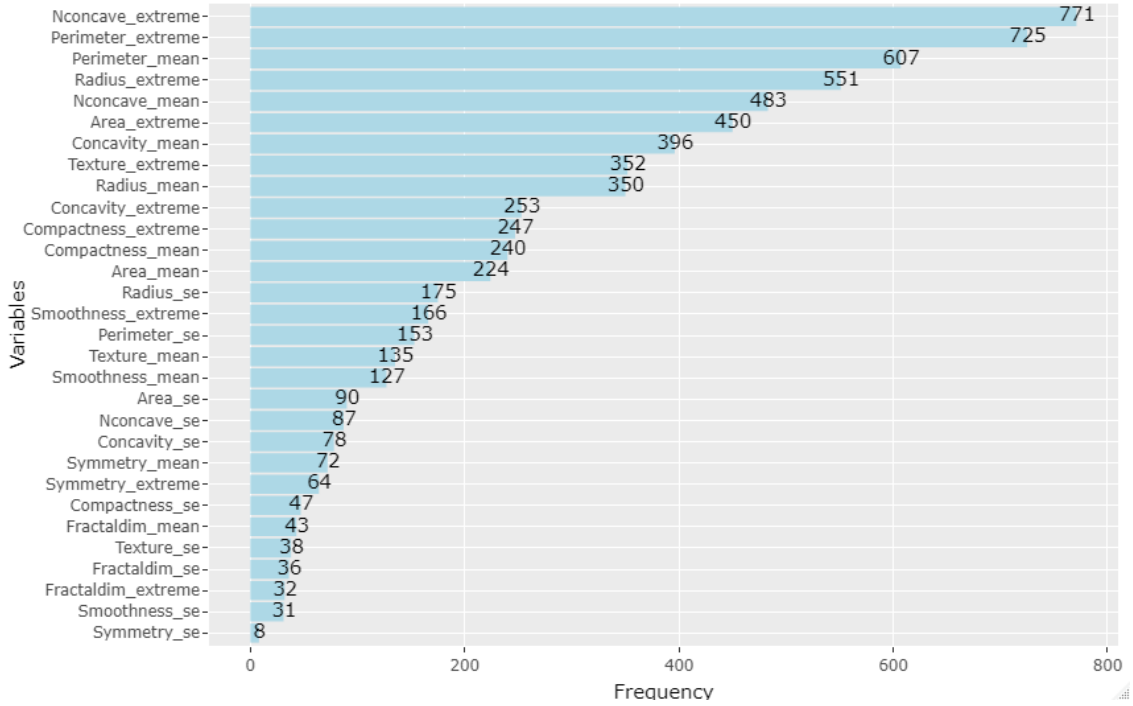



Figure 3.37: Frequency of number of variables selected by the greedy search algorithm for contaminated Wisconsin breast cancer data

In the contaminated diagnostic Wisconsin breast cancer dataset, it is evident that a model constructed using "selected variables" outperforms a model built with "all variables" in terms of class correct classification rates. However, the difference in contamination sensitivity between the two models is not substantial; the model incorporating variables obtained through greedy search exhibits slightly lower sensitivity (see Figure 3.38).

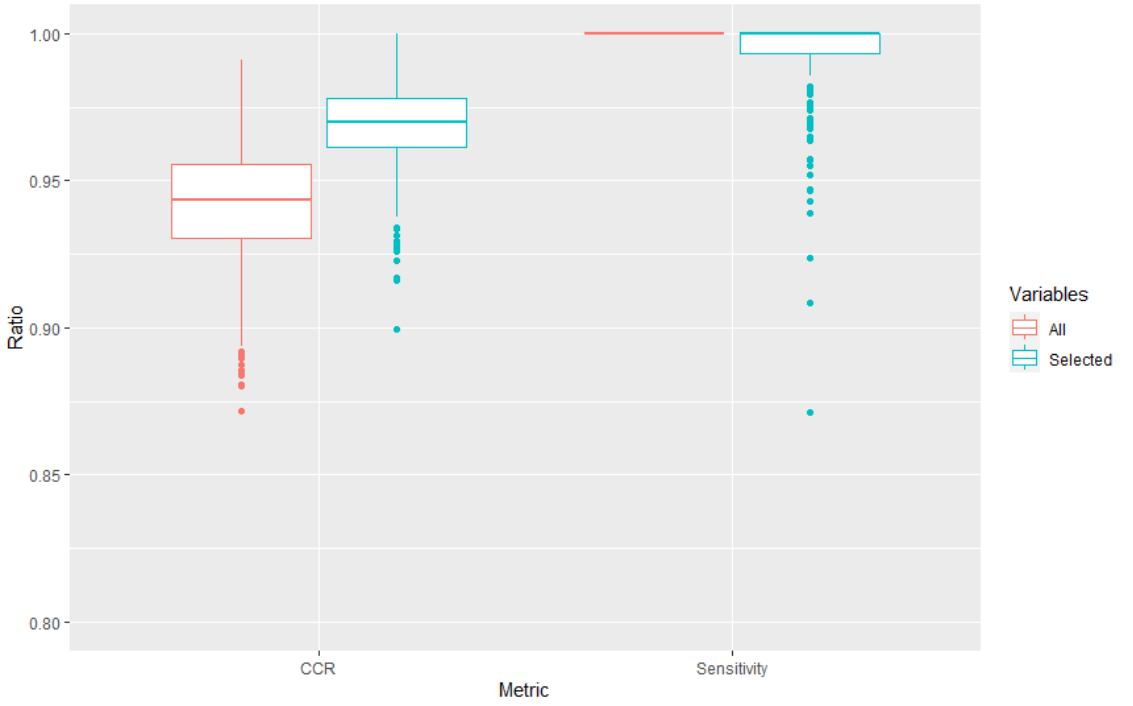



Figure 3.38: Class correct classification rate and contamination sensitivity by variable subset in the test Wisconsin breast cancer data

The observation indicates that changing the proportion of uncontaminated samples has little to no effect on CCR. This is evidenced by the positive difference in CCR between the model with variable selection and the one without, regardless of whether the proportion of non-contaminated samples remains consistent across classes. Additionally, in Figure 3.39 when comparing the differences in test contamination sensitivity between building models with "selected variables" and those using "all variables", the differences are small. Notably, they are marginally negative regardless of whether the percentage of non-contaminated observations is equal or not across classes.

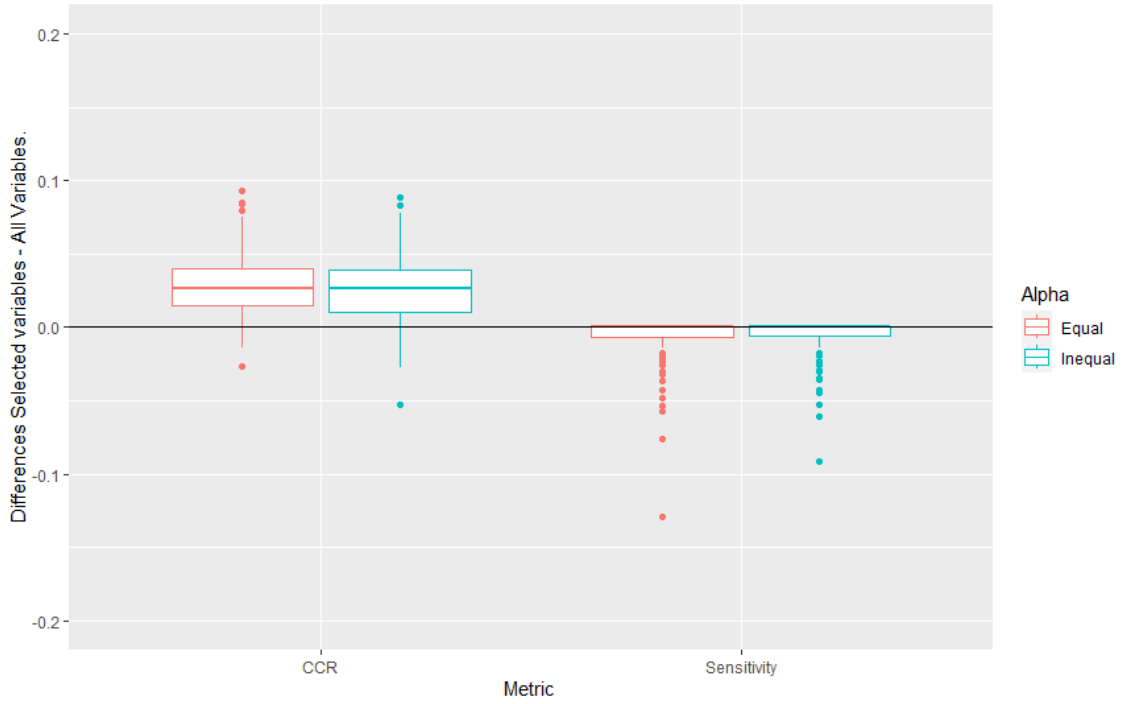



Figure 3.39: Difference in correct classification rate and sensitivity between selected variables for contaminated diagnostic Wisconsin breast cancer data varying parameter $\alpha$

In Figure 3.40, it is possible to observe that maintaining $\alpha$ constant while allowing $\eta$ to vary seems to have no noticeable effect on the differences in test class correct classification rate of models built with the "selected variables" and those including "all variables". These differences persist positively and unchanged. However, looking at the differences in test contamination sensitivity between using the subset of "selected variables" and "all variables", the differences are slightly negative. This means that there is a minimal improvement in identifying contaminated observations when the subset of "all variables" is used instead of the "selected variables" subset regardless whether the inflation factor is equal or unequal across classes. In this case the recommendation is to use the "selected variables" subset since the gains of including "all variables" subset are minimal.

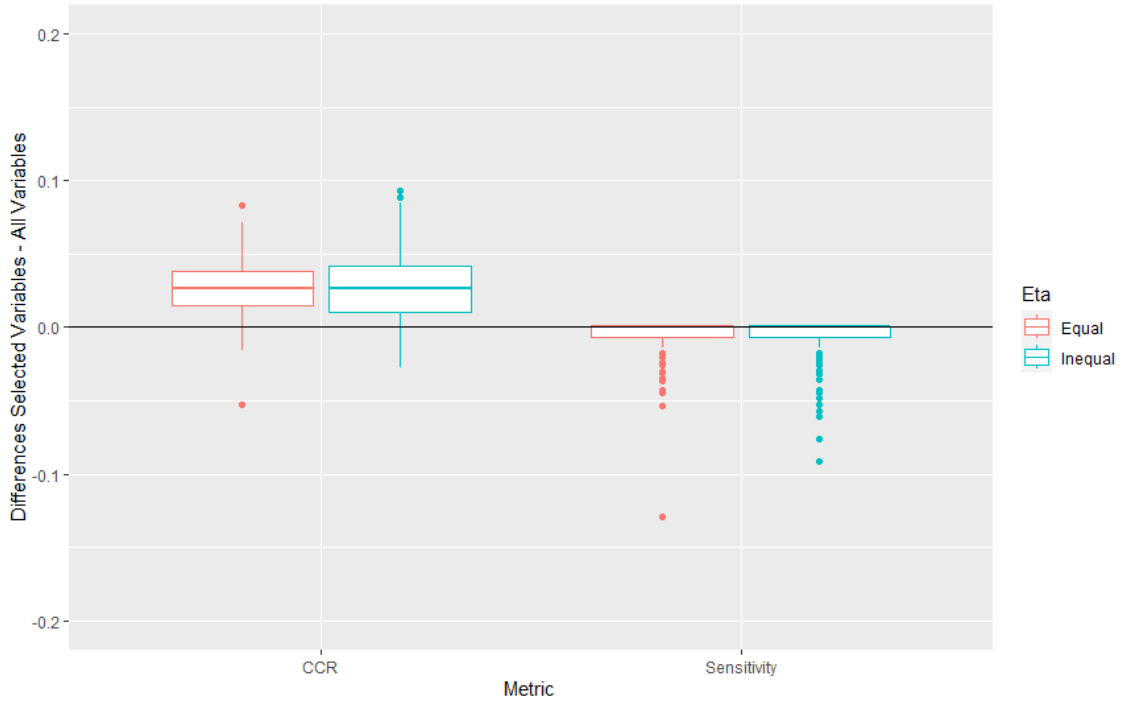



Figure 3.40: Difference in correct classification rate and sensitivity between selected variables and all variables for contaminated diagnostic Wisconsin breast cancer data varying parameter $\eta$

## 3.8 Discussion

The proposed approach integrates a supervised mixture of contaminated Gaussian models with a variable or feature selection method. It transforms the variable selection issue into a model selection challenge by employing a combination of contaminated Gaussian mixtures and the greedy search algorithm, with a Bayesian information criterion to choose the covariance structure that best fit the training data and the correct classification rate serving as the metric on the test set for variable selection. The strategy of testing the model on unseen data is a recommend practice to prevent overfitting and accurately assess model performance (Guyon and Elisseeff, 2003; Kohavi et al., 1995; Raschka and Mirjalili, 2019).

To compare a mixture of contaminated Gaussian with and without variable selection, the simulated data sets were generated with the number of observations necessary to fit the latter model. Hence, one benefit of the former method is that extends the application of a mixture of contaminated Gaussian models to cases where the number of observations

might be smaller than the number of variables, and to scenarios where there are numerous non-informative variables. The reason for this is that in forward selection, the relevant variables are incorporated into the model progressively, helping to potentially reduce the dimensionality of the data. Guyon and Elisseeff (2003) mention other benefits of variable selection such as data visualization and improved prediction performance.

Looking at the results of applying the proposed method to real data that have been contaminated to create plasmode datasets, it is found that in the case of crabs data, models made up of "selected variables" perform slightly better in correct classification rate regardless of values of percentage of non-contaminated observations $\alpha$ and amount of contamination $\eta$, but suffer a slight deterioration in their ability to identify contaminated samples (see Figures 3.24-3.25). The results confirm what we observed in the simulated cases, where the CCR improved when using the "selected variables". In this case, if the goal is to improve the CCR and reduce the complexity of the model, it is advisable to use models made up of "selected variables"; however, if the objective is to have a slight improvement in identifying contaminated samples, the model made up of all the variables should be used, provided there was contamination on all variables.

In the case of the wine data, the results suggest that there is no improvement when using the "selected variables" in CCR when $\alpha$ is modified, and an apparent improvement in accuracy when $\alpha$ is not the same across classes (see Figure 3.31). But looking in detail at this apparent improvement in accuracy and calculating sensitivity and specificity it is found that sensitivity deteriorates while specificity improves slightly (see Figure 3.32). The same behavior is repeated when $\eta$ varies (see Figure 3.33).

Although the wine data contains more variables than the crab data, the former has more classes and a smaller distance between the mean of its classes and as a result has an important overlap that affects the classification and also the identification of whether a sample is contaminated (see Figures 3.27). Regarding the covariance, there is a high negative correlation between $Color$ and $Hue$ variables, positive between $Flavanoids$ and $Hue$. These are variables that separate classes since they are included in the model in a significant number of simulations (see Figure 3.19b). The recommendation is similar to the wine data and is to use the "selected variables" if the objective is to simplify the model and obtain the same CCR levels, but if the objective is to identify contaminated samples

it is recommended to use the model that uses the "all variables" subset.

In the case of breast cancer diagnostic data, there is an improvement in the classification using the "selected variables" subset instead of the "all variables" subset regardless of whether alpha or eta are equal or not through all classes. However, although the accuracy plot suggests an improvement, looking at the specificity there is an improvement in identifying the uncontaminated classes, but there is no improvement or deterioration in sensitivity. The recommendation for this data set is to use the model made up of the "selected variables", whether the objective is the classification or identification of contaminated samples. The diagnostic data for breast cancer are different from the two previous ones since there are only two classes and the number of variables is much greater (30 variables). The results suggest that as the number of variables increases, it is possible to have variables that do not contain information about the class of the sample, so in a data set with a large number of variables it is better to use a model composed of "selected variables".

The method demonstrates effectiveness across various simulated and real datasets, achieving similar correct classification rates to those of a full model. Moreover, it consistently outperforms models formed solely with the true variables and, in many instances, models incorporating "all variables". Additionally, it expands the applicability of mixture of contaminated Gaussian models to scenarios with a larger dimensionality. The proposed method navigates the model space using a forward greedy search algorithm, ultimately converging to a local optimum solution. It is capable of handling large datasets. For instance, it underwent testing using 10 simulated datasets comprising 3 clusters, 3000 observations, and 100 variables (of which 3 were clustering variables). When executed on a laptop with 32 GB of memory and a 2.6 GHz processor, the unrestricted covariance model (VVV) required just under 6 hours of CPU time.

While the model exhibits reasonable performance in identifying contamination, the sensitivity of the model employing all variables is notably higher. This can be attributed to certain variables lacking group information but containing relevant contamination-related information. Additionally, the variable selection criterion primarily focused on accuracy improvement rather than sensitivity enhancement, which may explain why some variables

crucial for predicting contamination were overlooked. Therefore, while a model may excel in class prediction, it may not necessarily perform as well in identifying contamination in new observations, as evidenced by few instances.

# Chapter 4

# Variable selection with semi-supervised contaminated mixture of Gaussian models

## 4.1 Introduction

### 4.1.1 Previous work

Empirical evidence indicates that in certain scenarios, semi-supervised learning methods can leverage the abundance of unlabelled training data to enhance the performance of a learning task, thereby requiring fewer labelled training data to achieve a desired error bound. However, in other cases, unlabelled data do not appear to provide any assistance (Singh et al., 2008). Recent efforts have been made to theoretically understand whether unlabelled data are beneficial and under what circumstances. Castelli and Cover (1995, 1996) estimated the relative value of labelled and unlabelled data in a classification setting, concluding that labelled observations hold more value than unlabelled observations, However, unlabelled observations can still benefit a semi-supervised classification model, albeit to a lesser extent compared to labelled observations, provided the correct model assumptions are made. Yang and Priebe (2011) delve deeper into this topic and conclude that unlabelled observations contribute to a degradation in performance when the model is misspecified.

Finite mixture models have historically been employed for clustering, with initial con-

tributions by Edwards and Cavalli-Sforza (1965), Wolfie (1970), Binder (1978), and others. Additionally, McLachlan (1982), Celeux (1995), and Dasgupta and Raftery (1998) have further developed finite mixture models for clustering. The application of finite mixture models to address clustering problems, incorporating various family density distributions within the mixture, although predominately the Gaussian density distribution, is evident in the work of Fraley and Raftery (2002), McLachlan et al. (2003), Dean (2006), Bouveyron et al. (2007), McLachlan et al. (2007), McNicholas and Murphy (2008), Scrucca (2010), Morris et al. (2013), McNicholas et al. (2010), and several others. The use of finite mixture models of Gaussians is often known in the literature as model-based clustering. Model-based clustering methods can be adapted in the classification setting to incorporate the known group membership for a subset of observations and allow parameter estimation from both known and unknown group memberships, rather than only the known group observations as is the usual classification approach. These resulting methods are referred to as "partial classification" and fall within the realms of semi-supervised learning.

The concept of enhancing the performance of a classification model by combining unlabelled and labelled data is not novel and has been previously explored within the framework of finite mixture models by Hosmer (1973), Titterington (1976), Hosmer and Dick (1977), Smith and Makov (1978), Murray and Titterington (1978), Anderson (1979), McLachlan and Ganesalingam (1982), and Titterington et al. (1985). Andrews et al. (2011) noted that comparatively less work has been undertaken on semi-supervised model-based classification in contrast to model-based clustering. However, there has been recent interest in semi-supervised model-based classification, with Nigam et al. (1998) applying the Expectation-Maximization EM algorithm to classify text using labelled and unlabelled documents. Dean (2006) and McNicholas (2010) have extended Gaussian model-based clustering to the Gaussian mixture model-based semi-supervised framework.

### 4.1.2 New work

In the preceding chapter, the challenges posed by contaminated samples within classes and navigating high-dimensional datasets were addressed by employing a supervised mixture of contaminated Gaussian models, wrapped within a greedy search algorithm to both classify and reduce dimensionality. However, many real-world scenarios present extensive sets of unlabelled data, rendering manual labelling impractical due to its expense. To

mitigate this challenge, a small subset of samples is manually labelled. Consequently, a semi-supervised model, leveraging all available information from labelled and unlabelled data, is deemed more suitable for parameter estimation and predicting the class and contamination label of unlabelled observations. This chapter aims to extend the method proposed in Chapter 3 to incorporate variable selection for semi-supervised mixtures of contaminated Gaussian distributions.

In section 4.2.1 the general semi-supervised mixture of contaminated Gaussian models for continuous data is discussed. The proposed method of wrapping the semi-supervised mixture of contaminated Gaussian models with a forward greedy search algorithm is described in Section 4.2.2. Discussion of how to evaluate the performance of the model takes place in Section 4.2.3. Results on the performance of semi-supervised variable selection for a mixture of contaminated Gaussian models fitting various scenarios of the simulated datasets and a comparison with variable selection for supervised mixture of Gaussian distributions is presented in Section 4.3.10. An additional assessment of the model was conducted by fitting plasmode datasets in Sections 4.5 and 4.6. A discussion of the benefits and challenges of using this approach is given in section 4.7.

## 4.2 Methodology

### 4.2.1 Semi-supervised mixture of contaminated Gaussian models

Semi-supervised learning has been used in cases where where the number of labelled observations is small. The main idea is to use the most of the available information, labelled and unlabelled, when estimating classification model parameters. The training set of size $m$ is split into two categories: labelled observations $\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_k$ and unlabelled observations $\mathbf{z}_{k+1}, \ldots, \mathbf{z}_m$, where $z_i$ and $\mathbf{z}_i$ were defined in Section 2.7.1 and the contamination information of the observations $\boldsymbol{\nu}_i$ is missed for all $m$ observations. Semi-supervised learning combines clustering for unlabelled observations and discriminant analysis for labelled observations. The complete data-likelihood for a semi-supervised contaminated Gaussian model is given by:

$$L(\boldsymbol{\psi}) = \prod_{i=1}^{k}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{\tilde{z}_{ig}}$$
$$\prod_{i=k+1}^{m}\prod_{g=1}^{G}\left[\pi_g\left[\alpha_g N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\Big)\right]^{\nu_{ig}}\left[(1-\alpha_g)N\Big(\mathbf{x}_i|\boldsymbol{\mu}_g,\eta_g\boldsymbol{\Sigma}_g\Big)\right]^{(1-\nu_{ig})}\right]^{z_{ig}}, \quad (4.1)$$

Looking at Equation 4.1, it is possible to identify two components. The first $k$ observations are modeled with a discriminant analysis component while the next $m-k$ observations are modeled with a clustering component. The parameter estimation is obtained via ECM algorithm where in each iteration, the labelled data and unlabelled data is used to estimate the parameters and estimate labels for the unlabelled data. The estimates for $z_{ig}$ and $\nu_{ig}$ are computed using equations 2.24, 2.25.

Next we discuss the variable selection extension to this model.

### 4.2.2 Wrapping a semi-supervised mixture of contaminated Gaussian in a greedy search algorithm

The proposed approach introduced in Section 3.2.1, has been integrated in the semi-supervised version of the proposed model. Section 2.9.2 provided a detailed description of the greedy search algorithm. Here, Figure 4.1 shows a tailored forward greedy search algorithm for a semi-supervised model that uses contaminated mixtures of Gaussian distributions that incorporate labelled and unlabelled data to fit the model and parameter estimation for the case of continuous data.

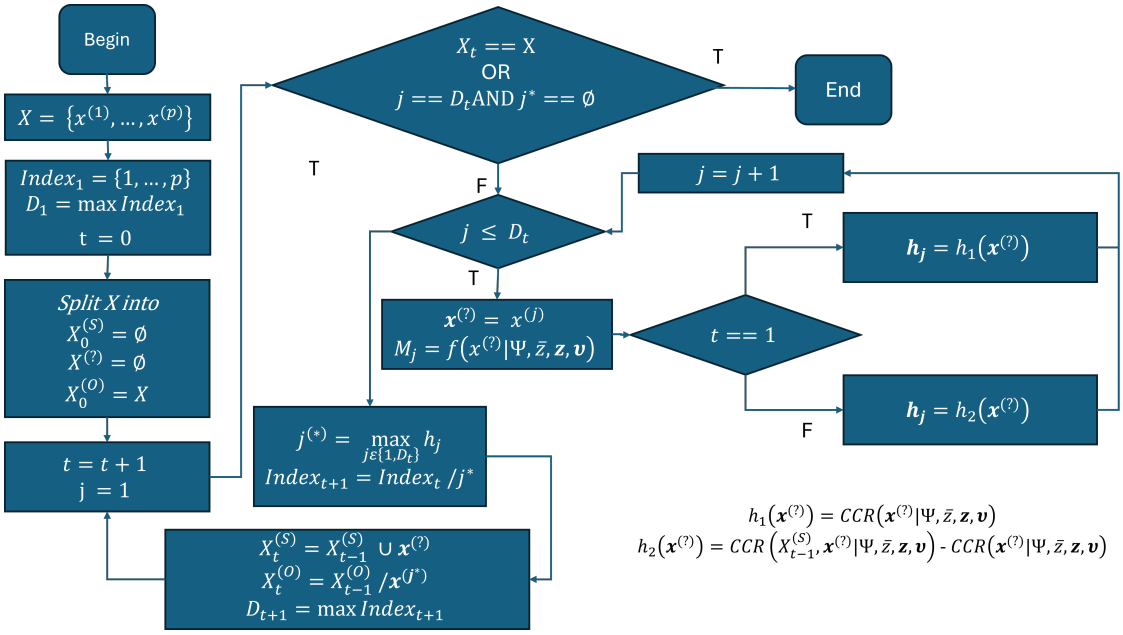



Figure 4.1: Tailored forward greedy search for a supervised contaminated mixtures of Gaussian distributions

The algorithm begins by partitioning all available variables into three subsets: one for the selected variables, initially empty; another for the proposed variable to include; also initially empty; and a third for variables not yet included in the model, initially containing all available variables. This third subset encompasses all available variables before the first step. The first step starts by checking the variables from the third subset for inclusion one at a time. Each of these variables constitutes a potential candidate model. All these models are fitted to the covariance structure models from the options in Table 2.2 using the ECM algorithm, and the covariance model providing the lowest BIC is chosen. The log-likelihood used to calculate the BIC includes the labeled and unlabelled data, as given in Equation 4.1. Once each of these models has been fitted to an appropriate covariance model, an E-step using Equation 2.23 is run to calculate label estimates for the unlabelled data in the test set. These predictions are used to calculate the class correct classification rate on the test set, and the variable yielding the highest correct classification rate in the test set is selected as the first variable for inclusion. Consequently, the first step concludes with adding the variable that provides the highest correct classification rate on the test set to the subset of selected variables and removing it from the subset containing all variables not included in the model.

In the general inclusion step, the variables in the subset of non-selected variables are as-

sessed individually in turn for inclusion. Each variable in the non-selected variables subset is evaluated individually for potential inclusion in the set of selected variables. The correct classification rate on the test set is calculated for both the proposed variable included and excluded from the variable set currently selected. The variable that results in the greatest improvement in test class correct classification rate is then added to the subset of selected variables and removed from the subset of non-selected variables.

The general inclusion step is iterated after the first step until either all variables in the non-selected variables subset have been rejected because they do not improve the test CCR over the currently selected set of variables or all variables have been included in the subset of selected variables. The main change in the adaptation of the variable selection wrapper for a semi-supervised mixture of contaminated Gaussian distributions lies in the log-likelihood that includes labelled and unlabelled data as can be seen in Equation 4.1 and the parameter estimates that use the equations of the ECM for a semi-supervised scenario incorporating labelled and unlabelled data in the conditional maximization steps.

### 4.2.3 Assessing a semi-supervised model using unlabelled observations in the training and test sets

In general, it is common practice to split the entire labelled dataset into two subsets. One subset is utilised in the training phase and the other subset in the test phase. Within the training set, there are both labelled and unlabelled data. In semi-supervised learning, an additional way to evaluate models is by taking into account the predicted labels in the training subset for those unlabelled data and comparing them with the true (but unused) labels associated with those observations. To illustrate how this can be used to evaluate a model a simulation of two balanced classes mapped in 5 dimensions with three separating variables was conducted.

In this scenario, 3000 observations were generated with the percentage of non-contaminated observations factor fixed for the first class to 80%, and for the second class to 90%. Additionally, the percentage of observations allocated to the training set was fixed to 75% and the variance inflation factor was fixed to 5 for the first class and 30 for the second class. The factors correlation structure $F_6$ was fixed to the case of having a strong correlation between separating variables and distance between mean classes $F_1$ varied at their respec-

tive levels and ten replicates were simulated for each scenario. Each of these replicates was composed of 3000 observations.

The training data was split into labelled and unlabelled data, ranging from 10% to 40% of the training data set to be labelled data (as often in practice, the amount of unlabelled data tends to exceed the labelled data). Table 4.1 illustrates the assessment of the model considering the prediction only of the unlabelled observations in the training set. The performance metrics train class correct classification rate, train class sensitivity, and train class specificity were calculated on the unlabelled observations in the training set at different percentages of unlabelled data in the training set of each replicate. In this scenario, the results suggest that the same performance obtained with 60% of unlabelled data and 40% of labelled data can be obtained with only 10% of labelled data and 90% of unlabelled data when you have a large number of observations overall.

Table 4.1: Average performance metrics on the unlabelled training data predictions at different percentages of unlabelled training data

| F2 | F6 | Type of prediction | Method | Performance metric | Percentage of Unlabelled Data | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | 60% | 70% | 80% | 90% |
| 2 | SCBSV | Class | Semi-Supervised | CCR | 0.75 | 0.75 | 0.75 | 0.73 |
| | | | | Sensitivity | 0.75 | 0.74 | 0.75 | 0.73 |
| | | | | Specificity | 0.75 | 0.74 | 0.75 | 0.73 |

## 4.3 Simulation studies

In a manner akin to the simulation studies carried out in Chapter 3 for variable selection on the supervised version of the proposed model, this section undertakes simulated studies to examine the performance of the variable selection semi-supervised model. The same simulated studies are conducted but with a modification: half of the observations in the training set are designated as unlabelled data. This adjustment means that, for each replication of scenarios wherein the proportion of observations allocated to the training set was adjusted to levels of 75% (equivalent to 2250 observations) and 85% (equivalent to 2550 observations), 1125 and 1275 observations, respectively, were treated as unlabelled data. Subsequent sections will describe the various simulation scenarios under consideration.

### 4.3.1 Simulations exploring the factor distance between class means

As previously mentioned, the distance between class means is identified as one of the influential factors affecting the performance of classification models. The differences in test data correct classification rate (CCR) for the models with different variable sets, with the factor of the distance between mean classes at different levels, while allowing the rest of the factors to vary, are depicted in Figure 4.2. It is evident that the differences between the "selected variables" and the true variables model, as well as between the "true variables" subset and "all variables", for the correct classification rate of test classes, are positive. This suggests that the model composed of the "selected variables" demonstrates slightly better performance in test class sensitivity and test class specificity across all levels. Upon assessing the different variable sets' models' performance in identifying contaminated observations, it becomes evident that the model incorporating "all variables" outperforms the other subsets of variables, with the "selected variables" yielding the second-highest test contamination sensitivity. Regarding contamination specificity, there is a minimal difference between the model containing the "selected variables" and including "all variables", with both sets generally surpassing the true variables model. In general, the highest differences between the "selected variables" and the true variables model, and the model including all the variables with the true model, were observed when there was an overlap between class means.

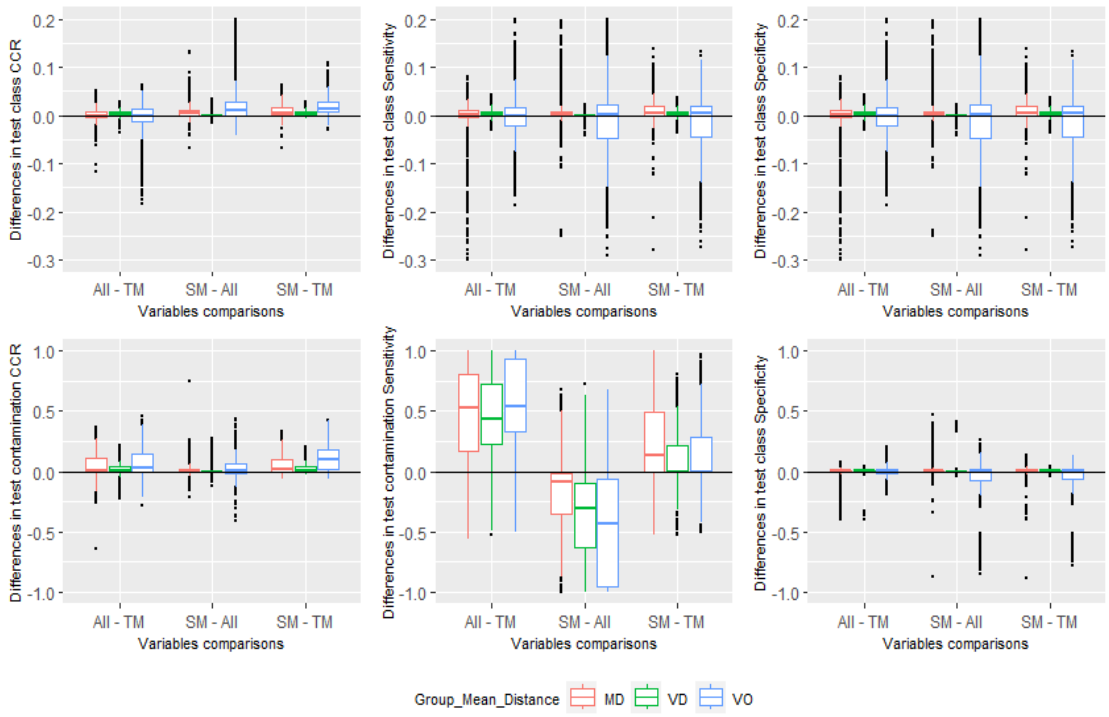



Figure 4.2: Boxplots of test CCR differences for models with different variable sets with different levels of distances between mean classes (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.2 Simulations exploring the factor number of classes

In Figure 4.3, the differences in test performance metrics for both classification and contamination detection among the three variable sets for scenarios where the factor number of classes was set at different levels and the remaining factors were allowed to vary are depicted. When assessing the models' ability to correctly allocate classes to new observations, similarities are observed among the three variable sets' models, with little differences in their test class correct classification rate, test class sensitivity, and test class specificity. Upon examining the performance metrics that measure contamination, it is evident that the model composed of the "selected variables" exhibits the best test class correct classification rate and the second-best performance in sensitivity, while the model including all the variables yields the highest sensitivity among the models. The differences in specificity among the three models are negligible, as they behave very similarly.

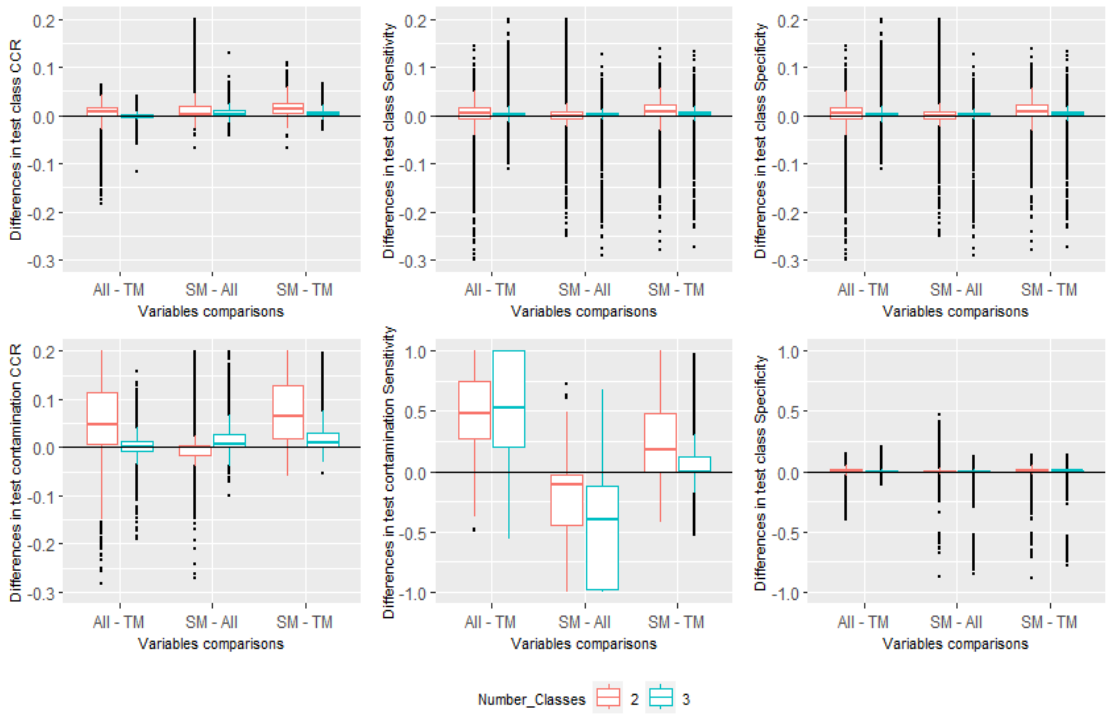



Figure 4.3: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of the number of classes (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.3 Simulations exploring the factor class proportion

The differences in performance metrics for the case where the class proportion factor was set at different levels while the other factors were allowed to vary are shown in Figure 4.4. Here, the effect of unbalanced classes is visible in the range of values that the differences in performance metrics took, since more extreme values are visible in the differences of the test class correct classification rate, test class sensitivity, and test class specificity. Additionally, the differences between the test contamination metrics of the model being compared illustrate a heavier impact on the "selected variables" and true variables than on including "all variables".

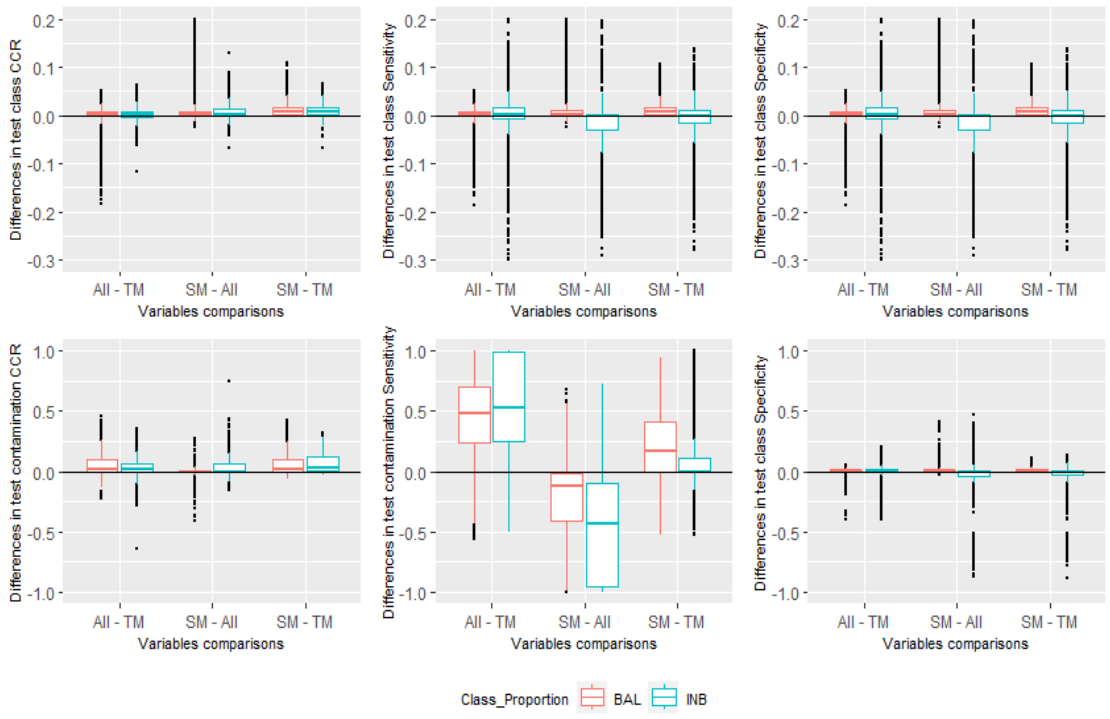



Figure 4.4: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of class proportion (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.4 Simulations exploring the factor number of variables

In Figure 4.5, the performance metrics of the variable sets' models under comparison illustrate differences in scenarios where the number of variables was set at different levels while other factors vary. It is evident that as the number of variables increases, so does the advantage of utilising variable selection in constructing a model, as opposed to solely relying on the true separating variables. Specifically, positive differences are observed when the "selected variables" were used compared to the other two variable sets' models, across metrics such as test class correct classification rate, test class sensitivity, test class specificity, and test contamination correct classification rate. This observation lends support to the notion of employing variable selection in high-dimensional data settings, notwithstanding a trade-off in sensitivity when a contamination pattern is present across all variables.

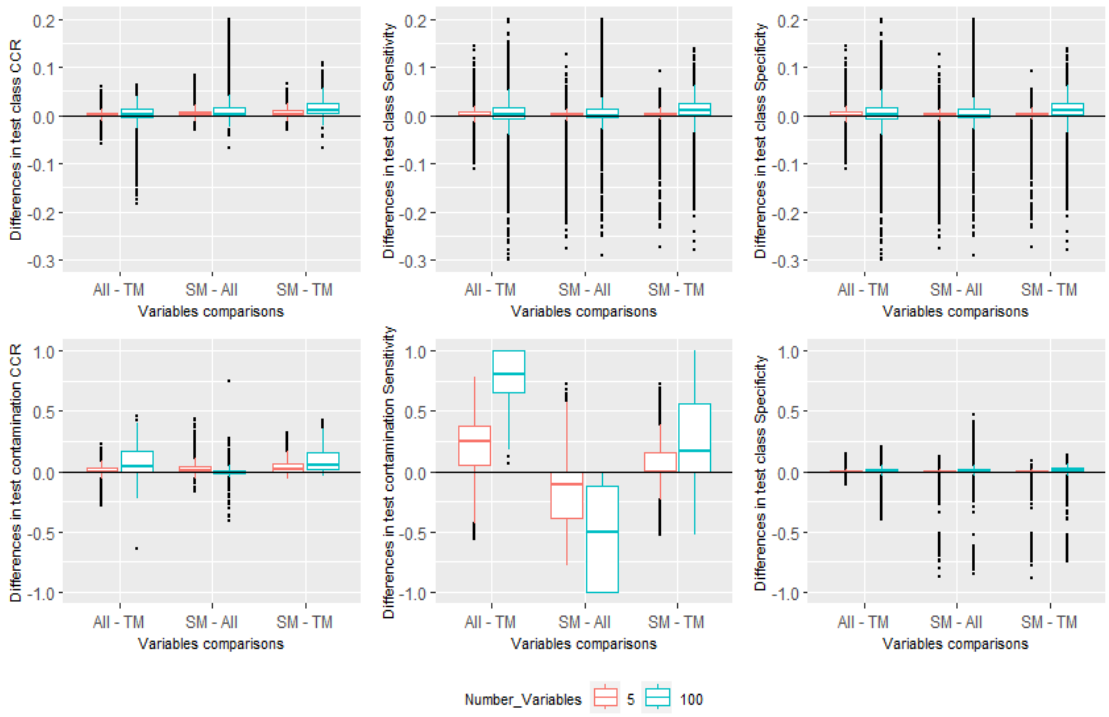



Figure 4.5: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of number of variables (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.5 Simulations exploring the factor percentage of samples used in training

The differences in performance metrics of the variable sets' models being compared for scenarios where the percentage of samples used in training was set at different levels and other factors were allowed to vary is illustrated in Figure 4.6. Although there are some extreme values in the performance metrics variable sets' model differences, it is noticeable that the central values regardless of the proportion of samples used in the training are positive which implies a slightly better performance of the "selected variables" in the test class correct classification rate, test class sensitivity, and test class specificity. In terms of detecting contaminated samples, the performance metrics variable sets' model differences show a slight improvement in favour of the "selected variables" and including "all variables" with an increase of 10% of the proportion of samples used in the training phase of these models.

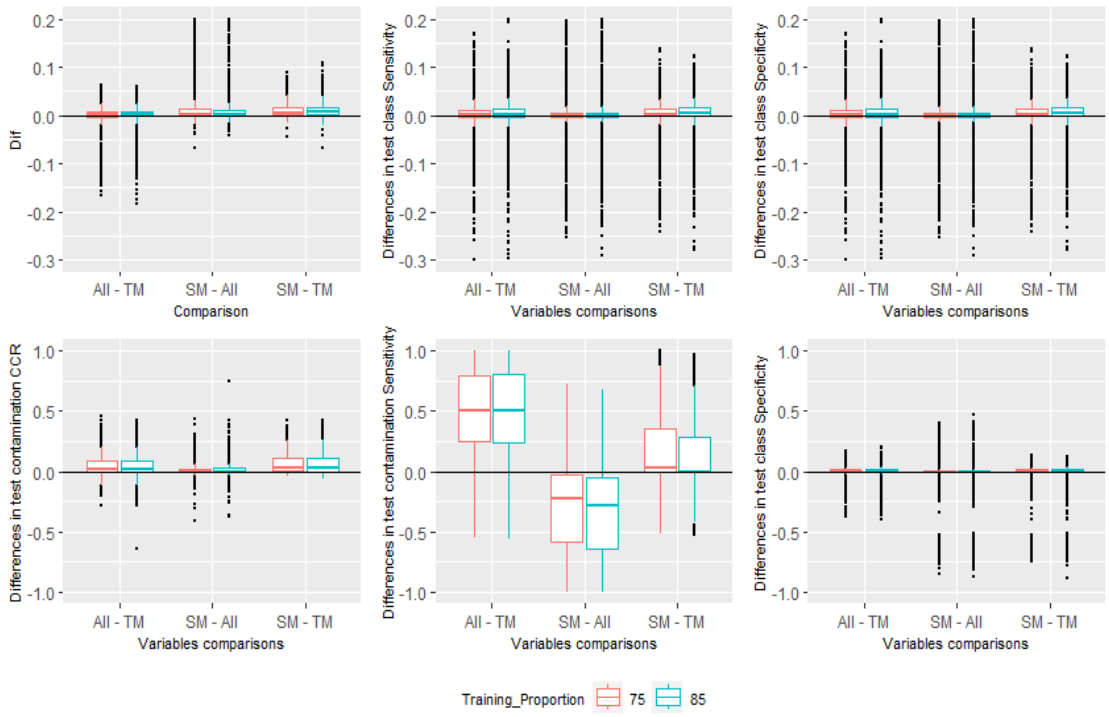



Figure 4.6: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of proportion of observations for training (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.6 Simulations exploring the factor correlation structure

The differences in performance metrics of the variable sets' models being compared for scenarios where the correlation structure factor used in training was set at different levels and others were allowed to vary are illustrated in Figure 4.7. Looking at the differences in the performance metrics of the variable sets' models, the results show positive differences in test class correct classification rate in favour of the "selected variables" when it was compared with the model including all the variables regardless of the correlation structure. Nevertheless, there is not much difference between variable sets' models in test class sensitivity and test class specificity regardless of the correlation structure. Assessing the differences in the performance metrics of the variable sets' models for contamination, it is noticeable the positive differences in test contamination correct classification rate and test contamination sensitivity are in favour of the "selected variables" when they were compared with the true variables model regardless of the correlation structure. These

positive differences are smaller when there is a strong correlation between separating variables which means that the "selected variables" might have not included the correlated separating variable if it does not improve the performance metric, but the true variables model certainly did.

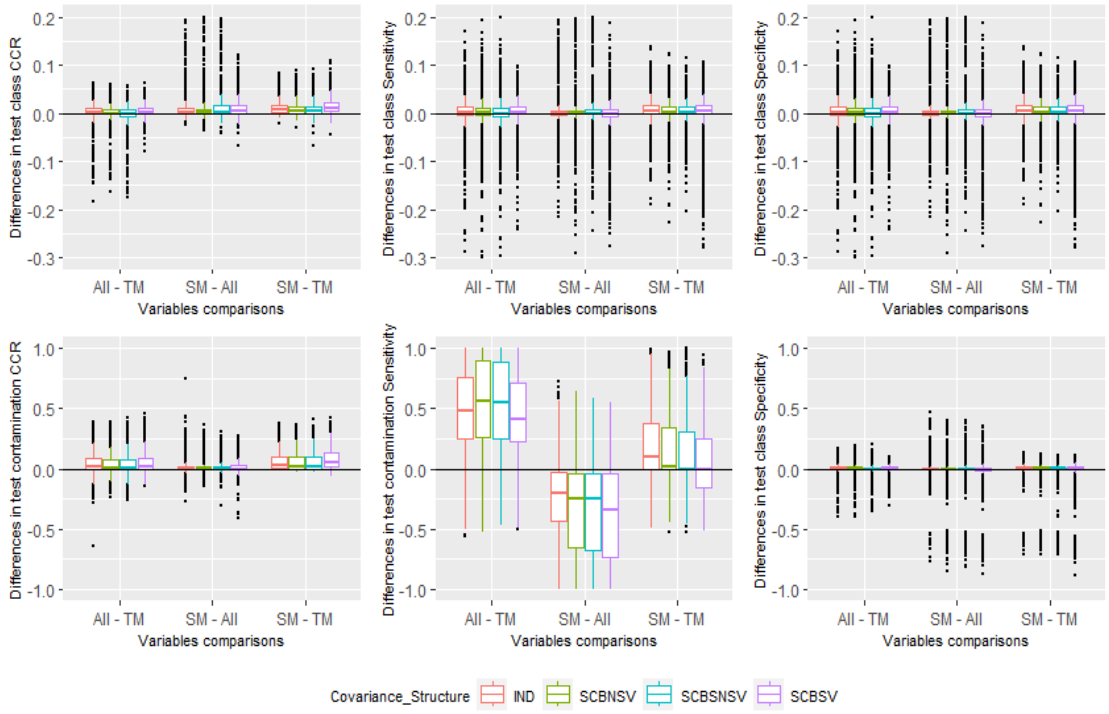



Figure 4.7: Boxplots of test CCR, sensitivity, and specificity differences for models with different variable sets with different levels of correlation structure fixed and other factors varied (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.7 Simulations exploring the factor number of separating variables

Figure 4.8 depicts the differences in performance metrics of the variable sets' models for scenarios where the number of separating variables was set at different levels and others were allowed to vary. The results show that the effect of adding an additional separating variable benefits the "selected variables" and the model including all the variables with a marginal increase in the test class correct classification rate and test contamination correct classification rate and it does not appear to have any impact in the performance metrics test class sensitivity and specificity and neither in test contamination specificity. Moreover, an increase in an additional separating variable on average terms reduces the

differences in test contamination sensitivity between the "selected variables" model and the model including all the variables. Although these differences still remain negative they are smaller than the scenario of not having them.

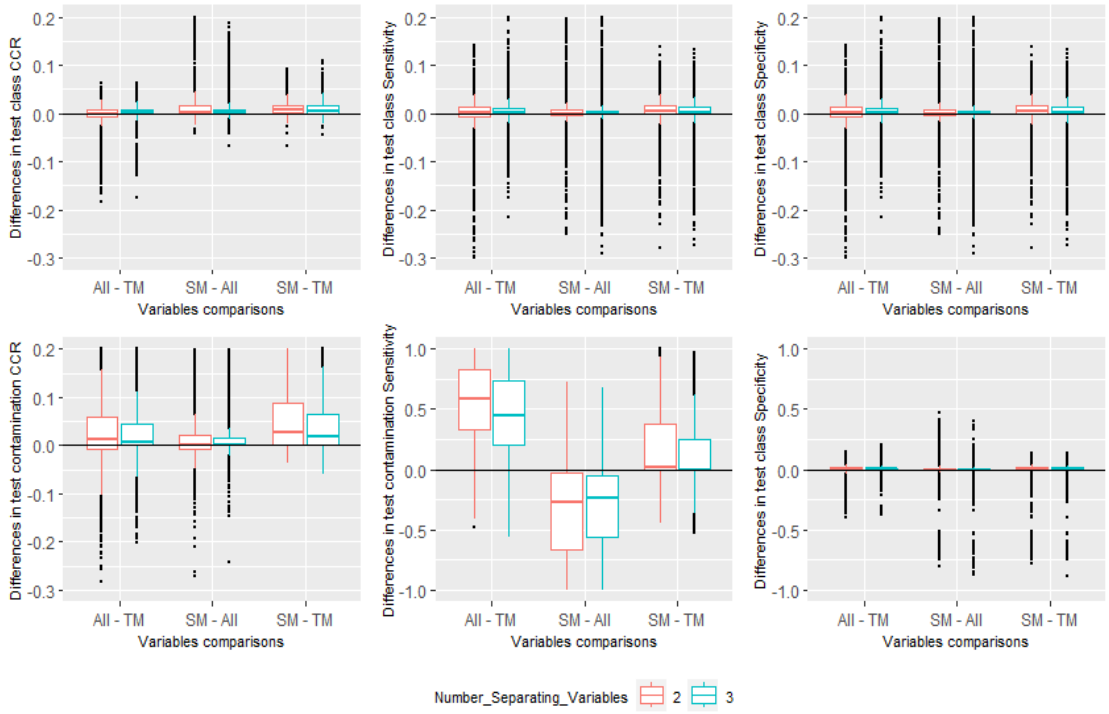



Figure 4.8: Boxplots of test CCR,, sensitivity, and specificity differences for models with different variable sets with different levels of number of separating variables (with other factors varied). First row's results are for classification performance, second row's for contamination performance.

### 4.3.8 Modelling mean of test class correct classification rate (CCR) by factors

It is pertinent to analyse the collective classification performance of the three sets of variables in different circumstances, namely if the performance fluctuates across different factor levels. The performance of each set is evaluated concerning the levels of each factor. It is expected to have between-scenarios variability that can be understood as the influence of the design factors across all three sets of variables under consideration, and within-scenarios variability as a product of having replicates for each scenario. As mentioned earlier different performance metrics from Section 2.3.2 were recorded at each simulated replicate of each scenario. However, there is a particular interest in two metrics the test

class correct classification and test contamination sensitivity. The interest in the former is due to the variable selection considered in the inclusion of a new variable, while the latter is a desired characteristic of the model addressing high-dimensional data.

A regression analysis with repeated measurements is conducted, focusing solely on the main effects. The model's intercept serves as a baseline for the test class correct classification rate. The baseline assumes only the true variables are in the model, a medium distance between class means, two balanced classes mapped in five dimensions, a proportion of 75% of the entire dataset used to train the model, independence between variables, and the existence of two variables that separate classes.

In Table 4.2 the ANOVA results suggest that sets of variables, the distance between mean classes, class proportion, correlation structure, and the number of separating variables are important predictors of the response variable - test class CCR -, while the number of total variables is also important but slightly less influential. The number of classes and proportion of samples assigned to training do not appear to impact the response variable.

Table 4.2: Anlysis of variance for test class CCR

| **Factor** | **numDF** | **F-value** | **p-value** |
|---|---|---|---|
| (Intercept) | 1 | 4394095 | $< .0001$ |
| Set_of_variables | 2 | 848 | $< .0001$ |
| Distance_between_mean_classes | 2 | 7723 | $< .0001$ |
| Number_of_classes | 1 | 0 | 0.6890 |
| Class_proportion | 1 | 903 | $< .0001$ |
| Number_of_variables | 14 | 15 | 0.0001 |
| Training_proportion | 1 | 0 | 0.6776 |
| Correlation_structure | 3 | 103 | $< .0001$ |
| Number_separating_variables | 1 | 558 | $< .0001$ |

There are three measures observed at each simulated replicate. For example, for a simulated replicated the test class correct classification rate is recorded for the set of variables "true"."selected", and "all". Consequently, these three different measurements taken for each individual are a source of variability that needs to be considered. These measure-

ments are referred to as repeated measurements and tend to be correlated (Laird and Ware, 1982). To represent these characteristics of the data a random effect is added with a compound symmetry to account for the correlation structure between measurements. Hence, the equation with the regression coefficients is expressed as

$$\begin{aligned} CCR_{class} = 0.8887203 &+ 0.0004558 \times VariablesAll + \\ &0.0135770 \times Variables_Selected + \\ &0.0276606 \times GroupMeanDistanceVD - \\ &0.1018257 \times GroupMeanDistanceVO + \\ &0.0002950 \times NumberClasses + \\ &0.0270154 \times ClassProportionINB + \\ &0.0000328 \times NumberVariables + \\ &0.0000759 \times TrainingProportion + \\ &0.0039634 \times CorrelationStructureSCBNSV + \\ &0.0016316 \times CorrelationStructureSCBSNSV - \\ &0.0164644 \times CorrelationStructureSCBSV + \\ &0.0212111 \times NumberSeparatingVariables \end{aligned} \quad (4.2)$$

The residuals plots provided in Appendix A show a deviation from normality, hence the p-values cannot be trusted. However, the regression coefficients still provide valuable information about the relationship between the factors and the test class's correct classification rate. It is clear that most of the factors appear to positively contribute to the test class CCR but having a very overlapping distance between mean classes and a strong correlation between separating variables. It might be seen that having unbalanced classes actually does, but to the detriment of the less representative class. Additionally, an additional separating variable and using variable selection offers the highest positive impact on improving the class correct classification rate. In the following section, the results of the variable selection procedure in the simulated datasets are presented.

### 4.3.9 Inclusion of informative and non-informative variables

As previously noted, a key benefit of simulated datasets lies in their known ground truth. Within various simulation scenarios, class separation is created by either two or three

variables. In real-world scenarios where these separating variables are unknown, a desirable characteristic of a variable selection method is its ability to identify and select them. To delve deeper into this assessment, this section presents the variables identified in simulation scenarios by the variable selection wrapped around the semi-supervised version of the contaminated mixtures model.

Table 4.3 presents a summary of performance of variable selection in the semi-supervised case for inclusion of separating variables and exclusion of non-informative ones in the scenario of two separating variables. For scenarios involving two separating variables, the average number of selected variables varies. When the total number of variables was 5, the average number of selected variables was approximately 3. However, when the number of variables was incremented to 100, the average number of selected variables rose to approximately 7.

Here, inclusion correctness is defined as the ratio of separating variables included in the final subset from the total number of separating variables. For instance, if there are 3 variables that create separation between classes, if the variable search only include one of this variables in the final subset, the inclusion correctness is $1/3$, if two of the separating variables makes their way to the final subset then the inclusion correctness is $2/3$. In terms of inclusion correctness, which assesses the inclusion of separating variables, the model demonstrates robust performance in including the relevant separating variables. Across both scenarios, denoted by $X_2$ and $X_4$, the algorithm achieves an inclusion correctness rate of 91% or higher, indicating its effectiveness in selecting the variables contributing to class separation. However, when the exclusion of non-informative variables is assessed, the performance varies. In the scenario with 5 separating variables, the algorithm achieves an exclusion correctness rate of 59%. In the scenario with 100 separating variables, the exclusion correctness rate substantially improves to 94%.

Overall the proposed method performs well in identifying and including the separating variables and excluding the non-informative ones.

The boxplots in Figure 4.9 illustrate the distribution of the number of variables included in the selected model. The results indicate that, on average, 3 variables were included in the model for classes mapped in 5 dimensions, while 7 variables were included for classes mapped in 100 dimensions. In terms of variability in the number of variables included, unsurprisingly there is greater variability when classes are mapped in 100 dimensions,

Table 4.3: Summary of the inclusion of separating variables and exclusion of non-informative ones in the scenario of two separating variables by the proposed semi-supervised model across varied factor levels

| Number of separating variables | Number of variables | Average number of selected variables | % time selected $X_2$ | % time selected $X_4$ | Inclusion correctness | Exclusion correctness |
|---|---|---|---|---|---|---|
| 2 | 5 | 3.07 | 92% | 91% | 94% | 59% |
| 2 | 100 | 7.10 | 80% | 80% | 85% | 94% |

attributable to a larger number of variables from which to choose. Additionally, It can be observed that in the scenario of classes mapped in 100 dimensions, occasionally the model expands the search to include around 23 variables. It is possible to see that in this example inclusion correctness can only take three values if the number of separating variables that found the way to the final subset was $0, 1$, or 2, then the inclusion correctness was $0, 1/2$, and $1/3$ respectively. Although most of the time the variable selection procedure include all the separating variables, there were a few times where it cannot identify any of the separating variables. These extreme cases where the variable search was not able to identify any of the separating variables occurs in some simulations where the classes were unbalanced.

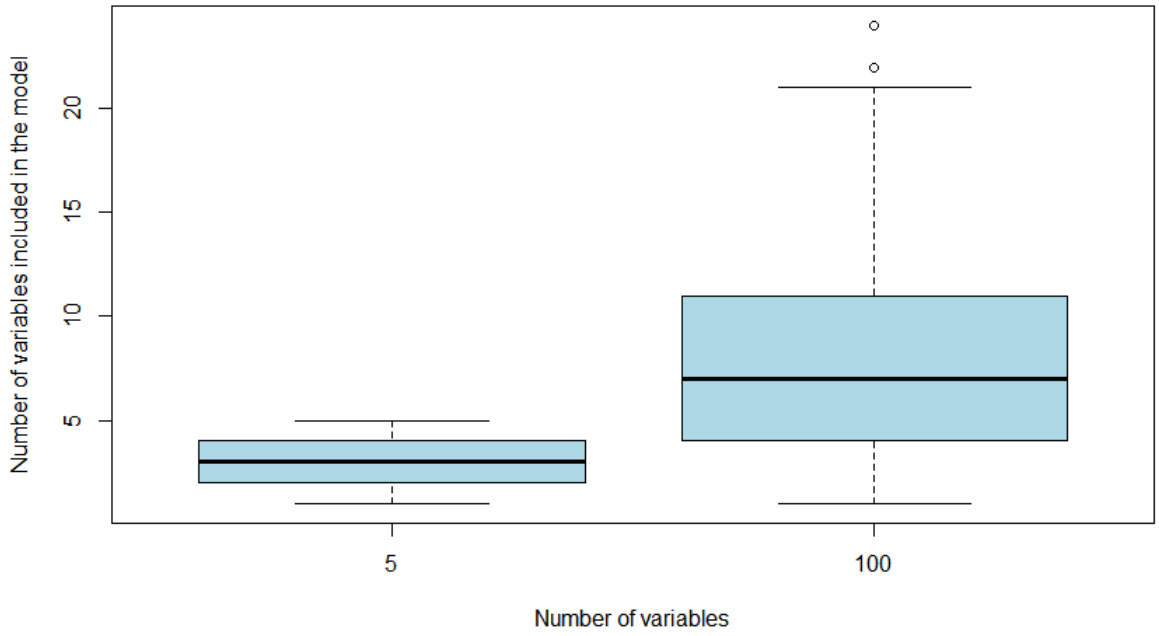



Figure 4.9: Boxplot of number of variables selected by the semi-supervised model across simulated datasets with two separating variables

Figure 4.10 presents the performance of the method identifying informative variables in the scenario where there are two separating variables. The figure suggests that almost always the proposed model was able to identify and include the variables that create separation between classes.

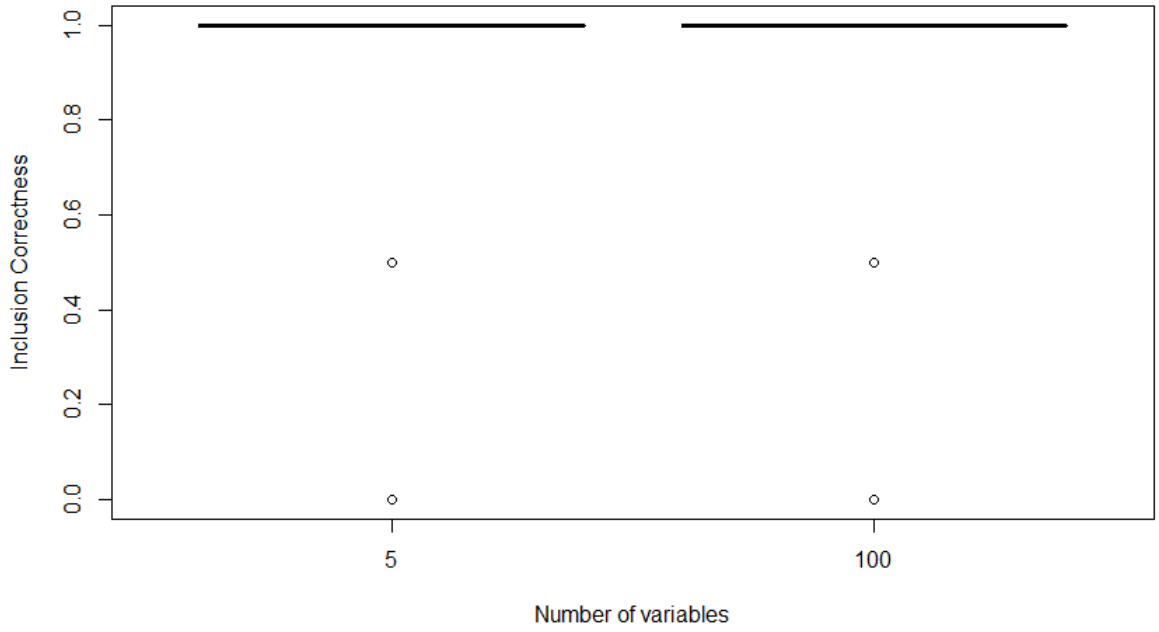



Figure 4.10: Boxplots of the inclusion correctness obtained by the semi-supervised model for scenarios with three separating variables

In a similar way Figure 4.11 presented the performance of excluding non-informative variables in the scenario where classes were mapped in 5 and 100 dimensions. Although in small datasets including one non-informative variables might have a big impact on exclusion correctness, the proposed method was able to identify and exclude most of the non-informative variable in both scenarios with 5 and 100 variables, showing a better performance in the scenario of classes mapped in a higher dimension.

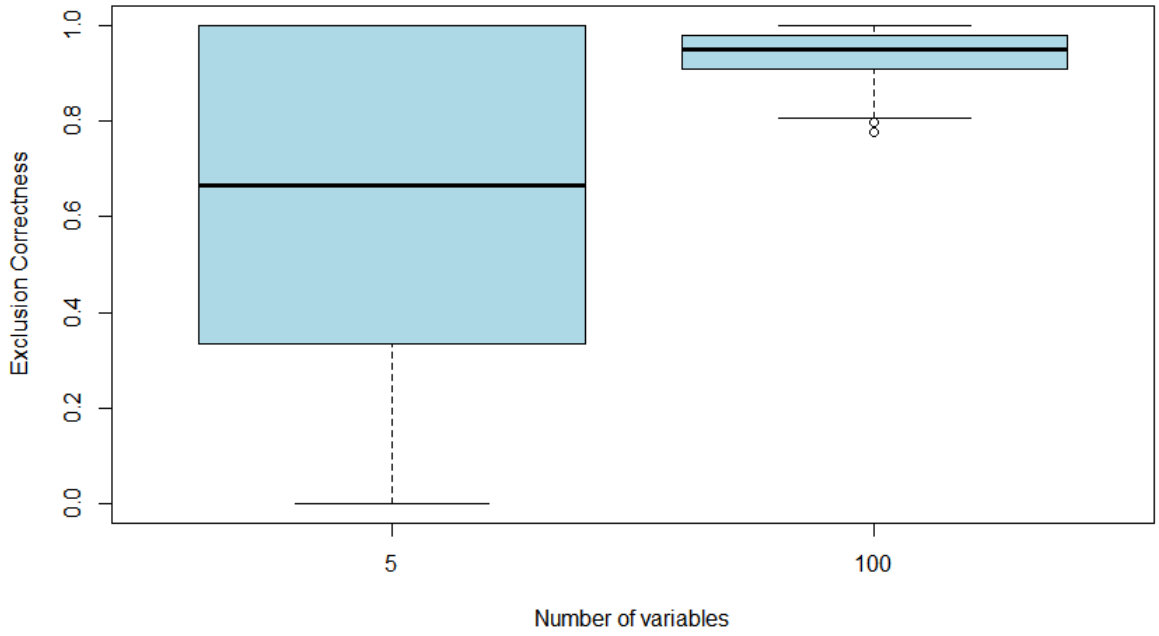



Figure 4.11: Boxplots of the exclusion correctness obtained by the semi-supervised model for scenarios with two separating variables

In Table 4.4 an evaluation of the performance of the proposed method in including informative and excluding non-informative variables in the scenario of three separating variables for classes mapped in 5 and 100 variables is presented. The average number of variables selected by the greedy search algorithm in scenarios where the classes were mapped in 5 and 100 dimensions were around 4 and 7 variables respectively, which is one additional variable in the case of classes mapped in 5 dimensions compared to the supervised version of the method in Chapter 3. Additionally, for classes mapped in 5 dimensions, the three separating variables, $X_2$, $X_4$, and $X_5$ were included in $89\%, 84\%$, and $94\%$ of the models, respectively. For class mapped in 100 dimensions, the three separating variables were included in the model $79\%$, $79\%$, and $80\%$ of times respectively. Evaluating the ability of the greedy search to include separating variables and exclude the non-separating variables, it is possible to see that for classes mapped in 5 dimensions, the variable search procedure includes $89\%$ of the separating variables and excludes $61\%$ of non-separating variables. In scenarios when classes are mapped in 100 dimensions, the algorithm search included $79\%$ of separating variables while excluding $95\%$ of non-separating variables.

Table 4.4: Summary of the inclusion of separating variables and exclusion of non-informative ones in the scenario of three separating variables by the greedy search algorithm across varied factor levels

| Number of separating variables | Number of variables | Average number of selected variables | % time selected $X_2$ | % time selected $X_4$ | % time selected $X_5$ | Inclusion correctness | Exclusion correctness |
|---|---|---|---|---|---|---|---|
| 3 | 5 | 3.46 | 89% | 84% | 94% | 89% | 61% |
| 3 | 100 | 7.17 | 79% | 79% | 80% | 79% | 95% |

In Figure 4.9 the distribution of the number of variables included in the model is presented. It shows that in cases where the number of variables was small (5 variables), the average number of variables included in the model varied between $3 - 4$ variables. Additionally, the variability of the number of variables included in the model grows with the increase of the pool of variables to choose from and the results reflect that even in that scenario, the number of variables included was on average less than 10% of the total available variables. It is also observed that occasionally the model ends up including between 20 and 25 variables. These might be the scenarios where a few variables provide a marginal gain in improving the test class correct classification rate.

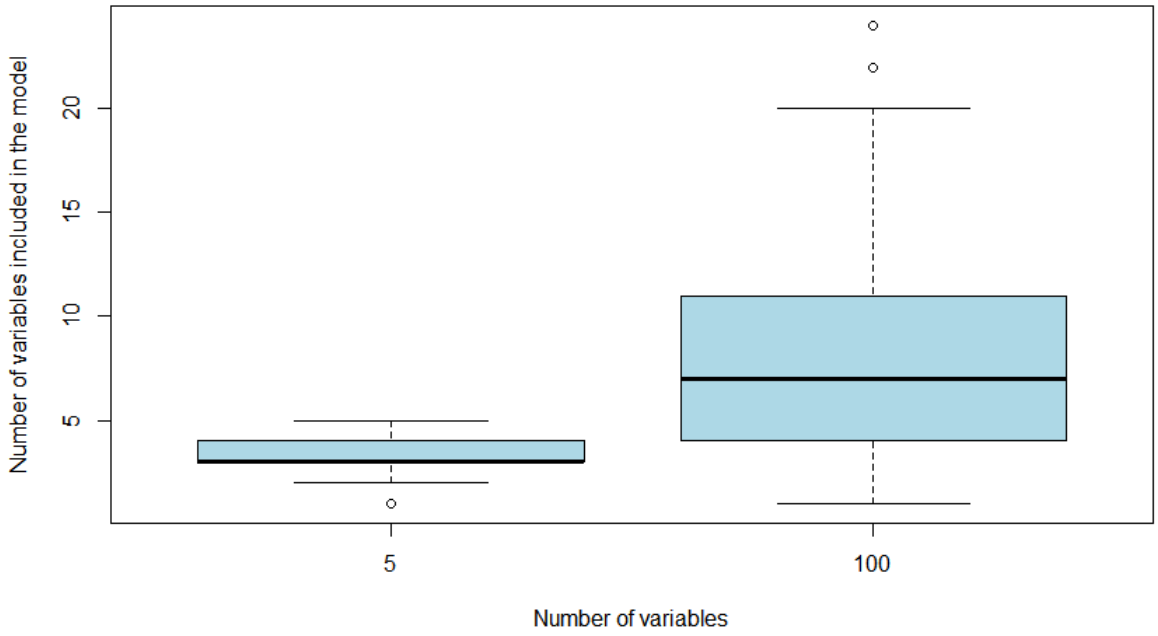



Figure 4.12: Boxplots of the number of variables selected by the semi-supervised model across simulated datasets with three separating variables

The distribution of inclusion correctness, as depicted in Figure 4.13, indicates that for classes mapped in 5 dimensions, the vast majority of the time, all the separating variables were included by the greedy search algorithm in the selected model and rarely conduct to cases where only one separating variable was included. While in scenarios where the classes were mapped in a higher dimension, the range of separating variables included in the selected model varied. In most instances, all of them were identified and included, but there were some scenarios only half of them were identified and included. There were exceptional cases, mainly scenarios with unbalanced classes, where the variable search failed to identify any of the separating variables.

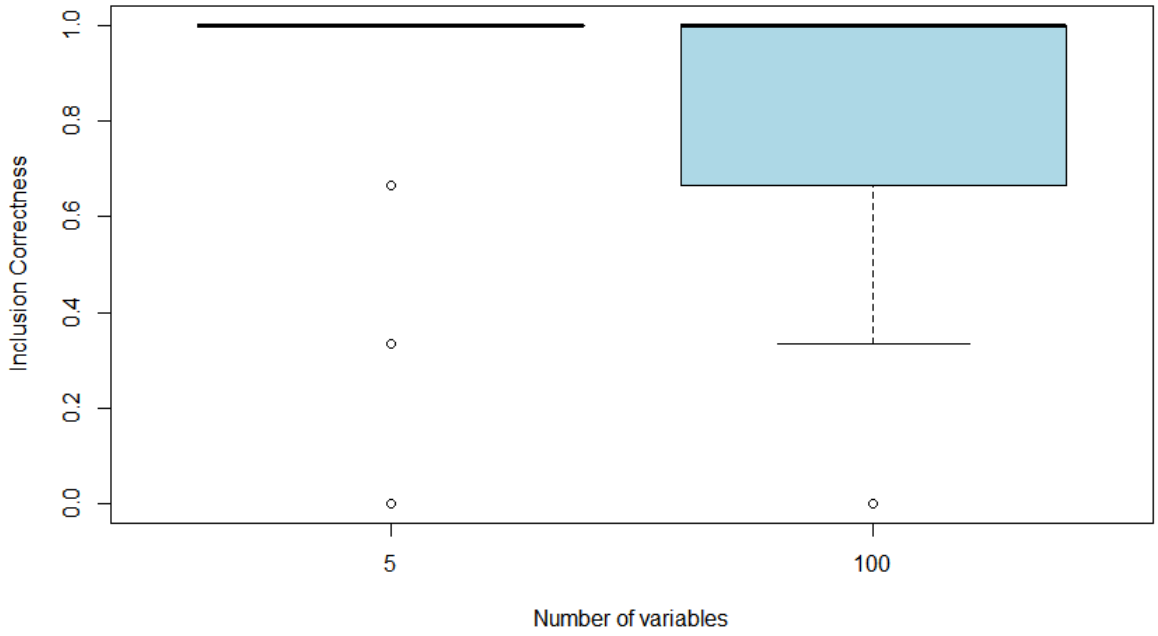



Figure 4.13: Boxplots of the inclusion correctness for scenarios with three separating variables

In Figure 4.14, the distribution of exclusion correctness is depicted for classes mapped in 5 and 100 dimensions. In scenarios where the classes were mapped in 5 dimensions, the variable search procedure predominantly excluded more than 50% of non-separating variables In scenarios where classes were mapped in a higher dimension, the variable search procedure discarded on average more than 90% of the non-separating variables and occasionally discarded between 80% to 90% of the non-separating variables in scenarios with unbalanced classes.

/

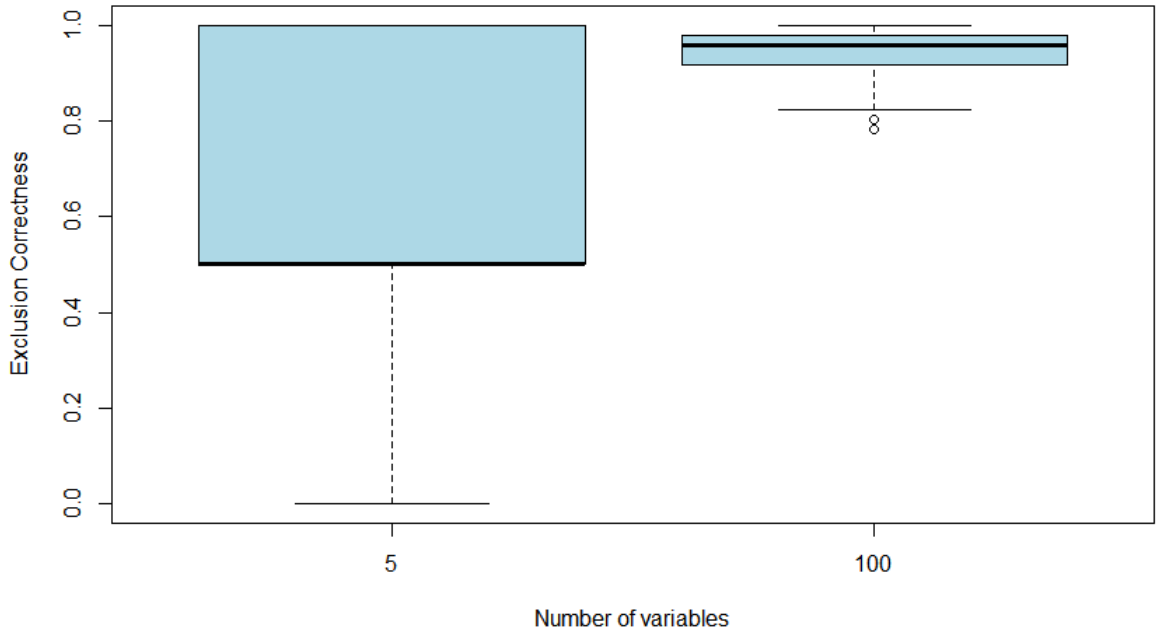



Figure 4.14: Variation of the exclusion correctness for scenarios with three separating variables

### 4.3.10 Comparison of semi-supervised and supervised learning

In this section, Table 4.5 showcases the mean correct classification rate for class variables and sensitivity for contamination. It compares the performance of variable selection between semi-supervised learning using a mixture of contaminated Gaussian distributions (SSL M-CG) with half of the data in the training set unlabelled, and supervised learning using a mixture of contaminated Gaussian distribution (SL M-CG). According to Steinley (2003), it is common to observe that overlapping between classes significantly influences classification models. The results indicate that overlapping has the most substantial impact on correct classification rate and sensitivity. Another influential factor is imbalanced classes, which have the second largest impact on sensitivity. Additionally, increasing the number of classes also affects sensitivity significantly. The semi-supervised model using "all variables" outperforms other models in terms of sensitivity, while models formed using "selected variables" consistently perform better in terms of correct classification rate.

Table 4.5: Test class correct classification rate means of SL and SSL methods by set of variables (V), distance between mean classes $F_1$, number of classes $F_2$, class proportion $F_3$, number of variables $F_5$, training proportion $F_6$, correlation structure $F_7$, number of separating variables $F_{10}$.

| Factors | Levels | SSL M-CG using true variables | SSL M-CG using selected variables | SSL M-CG using all variables | SL M-CG using selected variables |
|---|---|---|---|---|---|
| V | – | 0.94 | 0.95 | 0.96 | 0.97* |
| | – | | | | |
| | – | | | | |
| $F_1$ | Medium distance ($4.2\sigma$) | 0.96 | 0.97 | 0.97 | 0.97 |
| | Very distant ($8.5\sigma$) | 0.99 | 1.00 | 1.00 | 1.00 |
| | Very overlapping ($2.1\sigma$) | 0.87 | 0.89* | 0.87 | 0.88 |
| $F_2$ | 2 | 0.96 | 0.98 | 0.96 | 0.98 |
| | 3 | 0.97 | 0.97 | 0.97 | 0.98* |
| $F_3$ | Balanced | 0.96 | 0.97 | 0.96 | 0.97 |
| | Inbalanced | 0.96 | 0.97 | 0.97 | 0.97 |
| $F_4$ | 5 | 0.96 | 0.97 | 0.96 | 0.97 |
| | 100 | 0.96 | 0.98 | 0.97 | 0.98 |
| $F_5$ | 75 | 0.96 | 0.97 | 0.96 | 0.97 |
| | 85 | 0.96 | 0.97 | 0.97 | 0.97 |
| $F_6$ | IND | 0.97 | 0.98 | 0.97 | 0.98 |
| | SCBNSV | 0.97 | 0.98 | 0.97 | 0.98 |
| | SCBSNSV | 0.97 | 0.98 | 0.97 | 0.98 |
| | SCBSV | 0.94 | 0.96* | 0.94 | 0.95 |
| $F_9$ | 2 | 0.96 | 0.96 | 0.95 | 0.96 |
| | 3 | 0.97 | 0.98 | 0.98 | 0.98 |

Note: *Best performing method for the particular performance measure within each level of all factors.

Table 4.6: Test contamination sensitivity means of SL and SSL methods by set of variables (V), distance between mean classes $F_1$, number of classes $F_2$, class proportion $F_3$, number of variables $F_5$, training proportion $F_6$, correlation structure $F_7$, number of separating variables $F_{10}$.

| Factors | Levels | SSL M-CG using true variables | SSL M-CG using selected variables | SSL M-CG using all variables | SL M-CG using selected variables |
|---|---|---|---|---|---|
| V | – | 0.32 | 0.53 | 0.80* | *0.97 0.58 |
| | – | | | | |
| | – | | | | |
| $F_1$ | Medium distance ($4.2\sigma$) | 0.31 | 0.69 | 0.77* | 0.66 |
| | Very distant ($8.5\sigma$) | 0.37 | 0.51 | 1.00* | 0.53 |
| | Very overlapping ($2.1\sigma$) | 0.28 | 0.55 | 0.71* | 0.58 |
| $F_2$ | 2 | 0.39 | 0.67 | 0.89* | 0.66 |
| | 3 | 0.34 | 0.49 | 1.00* | 0.50 |
| $F_3$ | Balanced | 0.39 | 0.69 | 0.78* | 0.71 |
| | Inbalanced | 0.22 | 0.30 | 0.64* | 0.07 |
| $F_4$ | 5 | 0.33 | 0.47 | 0.64* | 0.47 |
| | 100 | 0.38 | 0.73 | 1.00* | 0.74 |
| $F_5$ | 75 | 0.31 | 0.61 | 0.76* | 0.61 |
| | 85 | 0.32 | 0.56 | 0.75* | 0.56 |
| $F_6$ | IND | 0.37 | 0.65 | 0.80* | 0.63 |
| | SCBNSV | 0.28 | 0.57 | 0.73* | 0.51 |
| | SCBSNSV | 0.28 | 0.57 | 0.72* | 0.57 |
| | SCBSV | 0.34 | 0.49 | 0.76* | 0.51 |
| $F_9$ | 2 | 0.28 | 0.56 | 0.76* | 0.52 |
| | 3 | 0.44 | 0.60 | 0.74* | 0.61 |

Note: *Best performing method for the particular performance measure within each level of all factors.

## 4.4 Plasmode data sets

In this section the plasmode data sets introduced in Section 3.4 will be revisited to evaluate the performance of the semi-supervised version of the proposed method. As mentioned earlier, using plasmode datasets offers the advantage of capturing patterns found in real-world applications. Since the real datasets lacked confirmed contaminated observations, simulated contaminated observations were generated for each class of these datasets. This was achieved by simulating a mixture of contaminated normal distribution at various levels of non-contaminated observations, denoted by $\alpha$, and an inflation factor denoted by $\eta$. By varying these parameters, the aim is to observe how the model performs and responds to changes in contamination levels.

Semi-supervised methods offer an additional way of evaluation compared to their supervised counterpart. This evaluation is based on the training subset. There are different procedures to conduct this evaluation. Here, the labels of the classes in the plasmode dataset are already known. To use observations in the training set to assess the semi-supervised model, once the dataset is split into training and test, half of the training set is assumed to be unlabelled and the model is fitted. After, the model is fitted, the predictions for the unlabelled observations in the training and test set are used to calculate the performance metrics of the model. Subsequent sections will focus on the analysis of the crab and Wisconsin breast cancer datasets and will show the performance metrics calculated on both subsets.

## 4.5 Results for the crab data

The crab dataset analysed previously in Section 3.5.2 is revisited here. This dataset contains different measurements of the physical attributes of the crabs. The physical measurements are taken mainly from the carapace of males and females from the species red and blue.

An analysis is conducted over the subset of blue species to illustrate the application of the variable selection for a semi-supervised mixture of Gaussians in a classification problem. The blue species contains a balance class composed of 50 males and 50 females. For the analysis, the dataset composed of blue species was contaminated for both sexes.

The procedure for generating contaminated observations is the same as in Chapter 3, where for both sexes the sample mean and variance were computed and these parameters were plugged in each of the mixtures of contaminated Gaussian modelling each class along with the values at with $\alpha$ and $\eta$ were varied to generate the contaminated samples that were introduced in the original dataset. The number of non-contaminated observations in each class was controlled by the respective proportion of non-contaminated samples for each class. The dataset was split into 75% for training and 25% for testing. The next sections show the results of the analysis.

### 4.5.1 Overall results for all values of $\alpha$ and $\eta$

The comparison between a set of variables based on the average of the Correct Classification Rate (CCR), sensitivity, and specificity for both training and test set is presented in Table 4.7. The model is evaluated on the training and test subsets based on a set of variables: selected and all, where all means including all the variables in the model. The results show that the CCR is higher on the training and test when using the "selected variables" compared to using "all variables". The model using "all variables" only outperforms the one using the "selected variables" in the training contamination CCR. The training contamination correct classification rate is higher when using a model that includes all the variables (All) in the crab dataset (0.90) compared to the selected model (SM) data (0.86) as Table 4.7 shows. Additionally, including all the variables in the model harms it as the test class and test contamination CCR decreases slightly, with values of 0.86 for class in the training set, 0.85, and 0.67 for class and contamination in the test set.

There is a consistency in the performance of the models using the "selected variables" on both training and test sets, since using the "selected variables" consistently yields the higher CCR values compared to using "all variables". This suggests that incorporating only the "selected variables" in the model leads to higher test class and contamination CCR.

Table 4.7: Comparison between a set of variables in average correct classification rate crab data

| Metrics | Set of Variables | Training | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| CCR | Selected | 0.87 | 0.86 | 0.89 | 0.76 |
| | All | 0.86 | 0.90 | 0.85 | 0.67 |

Consequently, it can be seen in Figure 4.15 that the differences in CCR between the selected model and the model including all the variables in the crab dataset are positive for both test class CCR and test contamination CCR. Although there is a higher variation of CCR on the test set, it is also positive, supporting variable selection for this dataset.

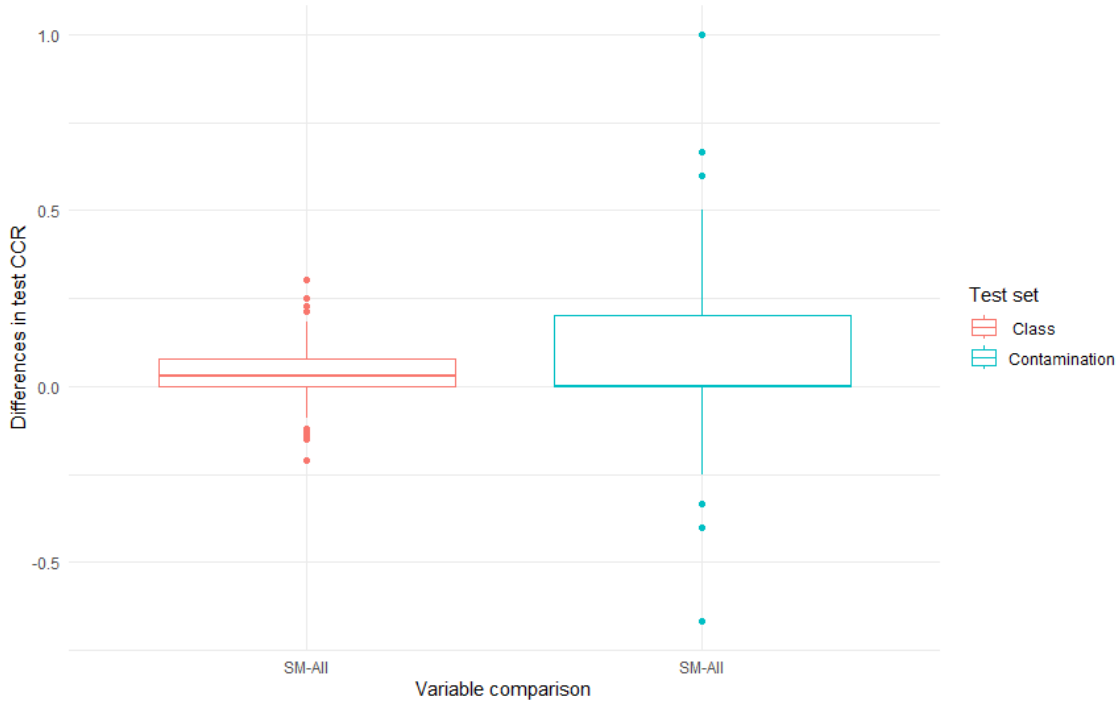



Figure 4.15: Differences in CCR by models on the crab test set.

In Table 4.8 a comparison of sensitivity between the "selected variables" and the "all variables" subsets is shown. The train class sensitivity seems comparable for the "selected" and "all variables". Nevertheless, there is a big difference in the train contamination sensitivity scenario where the "selected variables" achieve a sensitivity of 0.29 while using "all variables" produces a better performance with a sensitivity of 0.52.

In the test set, the sensitivity values show a similar pattern to the train set. For the class scenario, both selected and "all variables" exhibit similar sensitivity values, with the "selected variables" outperforming "all variables" at 0.90 compared to 0.85. However, in the task of identifying contaminated observations using "all variables" performs better than using only the "selected variables" with a sensitivity of 0.72 compared to 0.55 for the "selected variables".

Table 4.8: Comparison between sets of variables in average sensitivity crab data

| Metrics | Set of Variables | Train | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| Sensitivity | Selected | 0.87 | 0.29 | 0.90 | 0.55 |
| | All | 0.86 | 0.52 | 0.85 | 0.72 |

Looking at the differences in the test set produced by the comparison of the average sensitivities obtained by the "selected variables" and "all variables" subsets in Figure 4.16, it is noticeable that the differences are slightly positive for test class sensitivity. However, their differences in the test contamination sensitivity are negative with a higher level of uncertainty varying from 0 to approximately $-0.35$. These results imply that although there is a slight improvement in sensitivity in allocating a class to new observations when the "selected variables" are used, there is also a decrease in sensitivity in identifying contaminated observations.

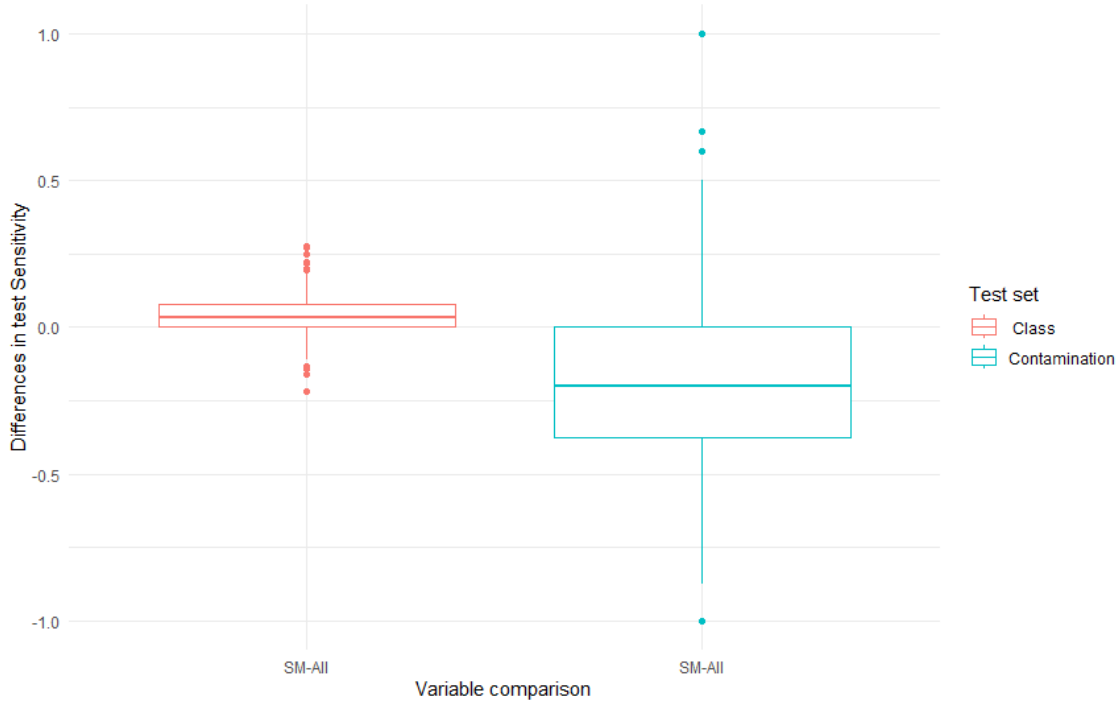



Figure 4.16: Differences in sensitivity by models on the crab test set.

The comparison of the average specificity between "selected" and "all" variables for both training and test subsets is shown in Table 4.9. In the training phase, the selected set of variables reached an average specificity of 0.87 for class prediction and 1.00 for contamination prediction while "all variables" subset reached an average of 0.86 for class prediction and 0.99 for contamination prediction. Similar to the training phase, the selected set of variables outperforms the "all variables" subset. Specifically, the average specificity is 0.92 for class prediction and 0.96 for contamination prediction using the "selected variables" subset. However, the performance of the "all variables" subset decreases slightly in the testing phase, with an average specificity of 0.85 for class prediction and 0.92 for contamination prediction. Overall, these findings suggest that the Selected set of variables generally leads to better specificity in both the training and testing phases compared to the "all variables" subset.

Table 4.9: Comparison between sets of variables in average specificity crab data

| Metrics | Set of Variables | Train | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| Specificity | Selected | 0.87 | 1.00 | 0.90 | 0.96 |
| | All | 0.86 | 0.99 | 0.85 | 0.92 |

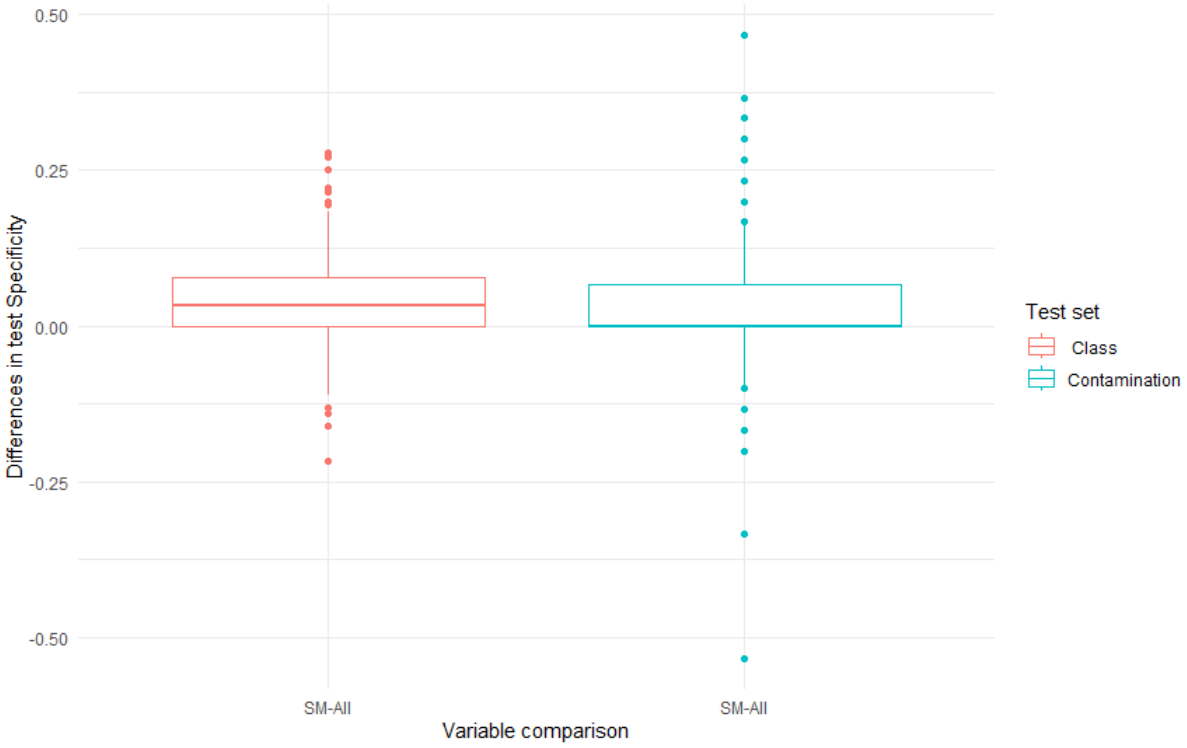



Figure 4.17: Differences in specificity by models on the crab test set.

### 4.5.2 Results for differing versus some $\alpha$

The percentage of non-contaminated observations in each sex varied at levels $75\%, 80\%$, and $85\%$ to assess if the performance of the models was affected. The differences in the performance metrics of the models for the class prediction and contamination identification were registered.

In Figure 4.18 the differences in CCR for both set of variables are shown. It can be seen that the differences in CCR between both models are positive in class prediction and identification of contaminated observations regardless of the values of $\alpha$'s. This result

suggests using variable selection to build a classification model. Additionally, a lower variability in CCR was observed in the identification of contamination compared to the class prediction scenario mainly when $\alpha$'s took the value of 0.85. It is also observed that occasionally, there were situations where the "selected variables" outperformed by far using all the variables producing some outliers, and also some cases where using "all variables" gave a higher test class's correct classification rate. In the test contamination correct classification rate, there were a few scenarios where using the "selected variables" produced a poor performance in both classes. These results were observed more frequently when $\alpha$'s took the values of 0.75 and 0.80.

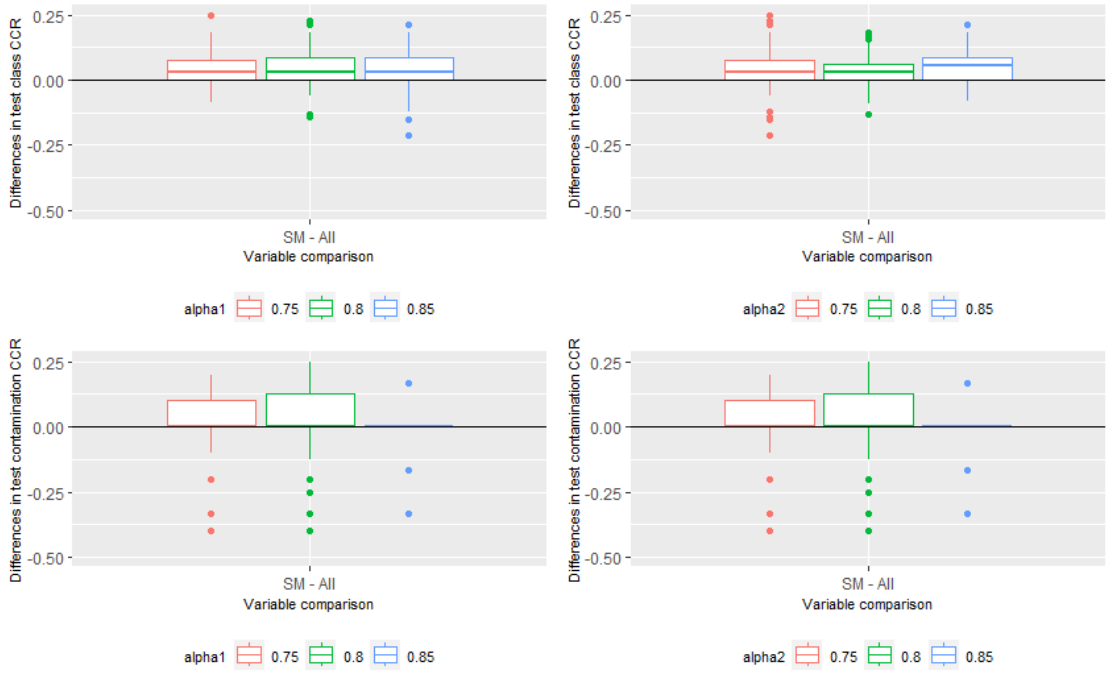



Figure 4.18: Differences in average CCR by subsets of variables on the crab test set at each level of $\alpha$.

The differences in sensitivity in class prediction and contamination identification are plotted in Figure 4.19. The effect of varying $\alpha$ appears not to have any effect on sensitivity in the context of class label prediction observing the first row of the plot. Observing the effect of varying $\alpha$'s on the contamination scenario it is noticeable that all the differences are negative, suggesting a better performance including all the variables. Also, the variability of the differences increases in the contamination scenario compared to the classification scenario.

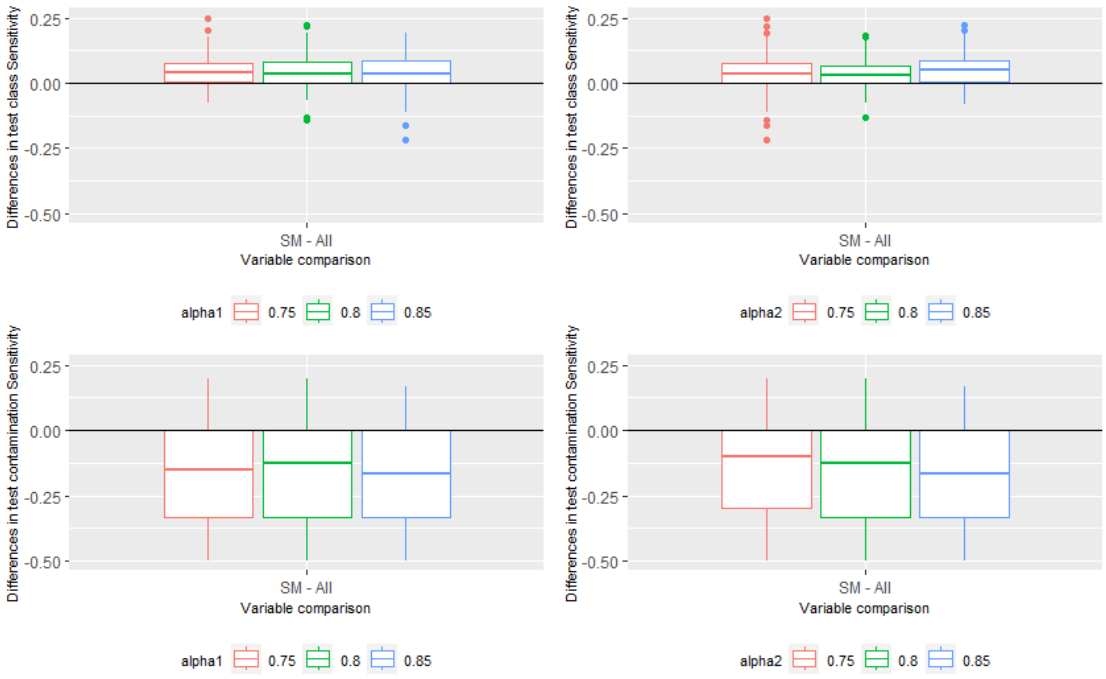



Figure 4.19: Differences in sensitivity by models on the crab test set at each level of $\alpha$.

Looking at Figure 4.20 it is visible that the differences between both sets of variables are small in the classification scenario regardless of the variation of $\alpha$'s in each class. In the test contamination specificity scenario, it is observed that when $\alpha$ was varied in the first group to 0.85 there was a decrease in the differences between using both sets of variables while the second class showed an increase in the differences in specificity for the two subsets of variables. These results suggest that there is a slight improvement in specificity using the "selected variables".

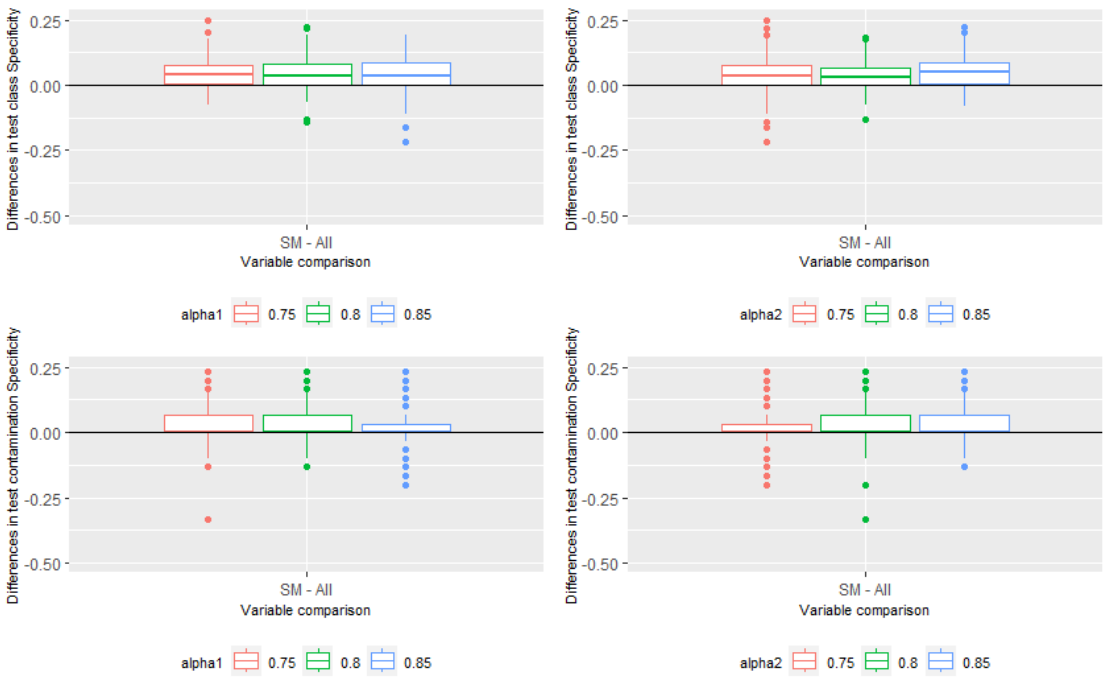



Figure 4.20: Differences in specificity by models on the crab test set at each level of $\alpha$.

### 4.5.3 Results for differing versus some $\eta$

The inflation factor $\eta$ varied at levels $5, 10$, and 15 to assess if the performance of the variable selection is affected. The differences in performance metrics for two sets of variables were recorded for classification and contamination performance on the crab test set.

In Figure 4.21 the differences in classification performance (first row of the plot) of CCR are positive regardless of the level of $\eta$'s on both groups. This implies that the "selected variables" outperform the set of "all variables". A similar pattern of positive differences regardless of the level of $\eta$'s is observed when looking at the contamination performance in the second row of the plot. Nevertheless, the variation of the differences decreases for the first group at the level of 15 and the number of outliers increases, while at levels 5 and 10 the average of the differences approaches 0 in the first class. For the second class the differences in CCR for $\eta_2$ at levels 5 and 10 almost vanish, but also show a slight increase in the number of outliers at those levels and keep positive at level 15.

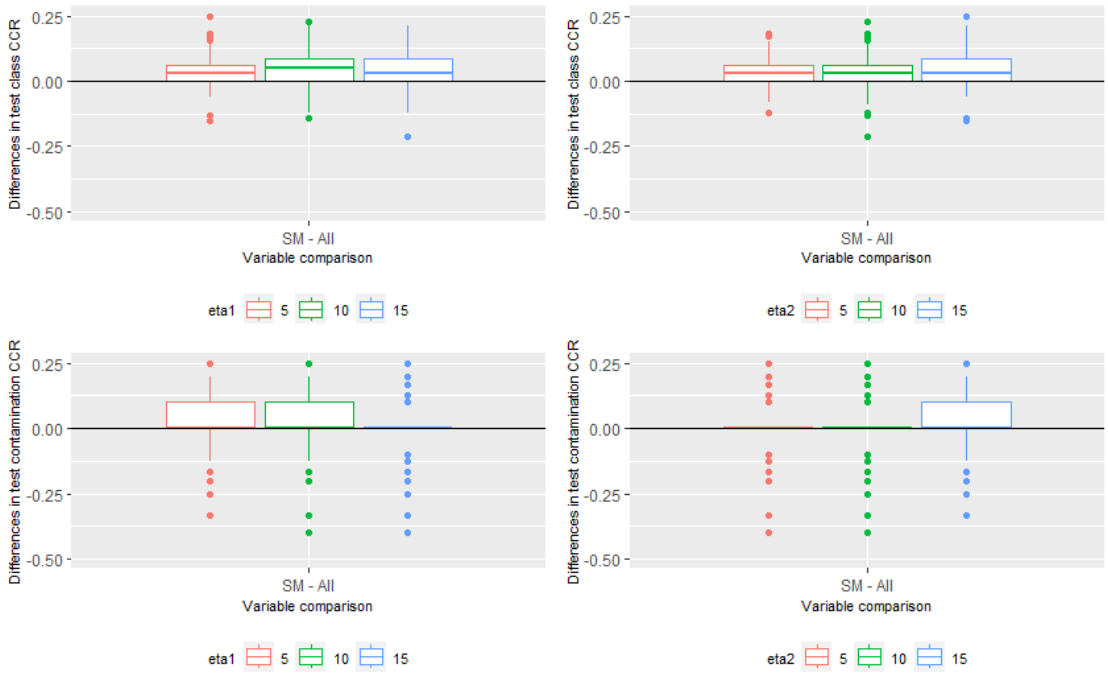



Figure 4.21: Differences in CCR by models on the test crab set at each level of $\eta$.

In Figure 4.22 positive differences are observed in the first row that correspond to classification performance. These positive differences seem not to be affected in both classes by any variation of $\eta$'s. Nevertheless, it is possible to notice a few extreme values where the "all variables" subset reached a higher sensitivity than the set of "selected variables" and a slight increase in sensitivity for the second class when $\eta$ increases. Looking at the second row, the contamination performance reveals that the differences are negative for both classes, indicating a better performance for the set of "all variables" at all levels of $\eta_1$ and $\eta_2$.

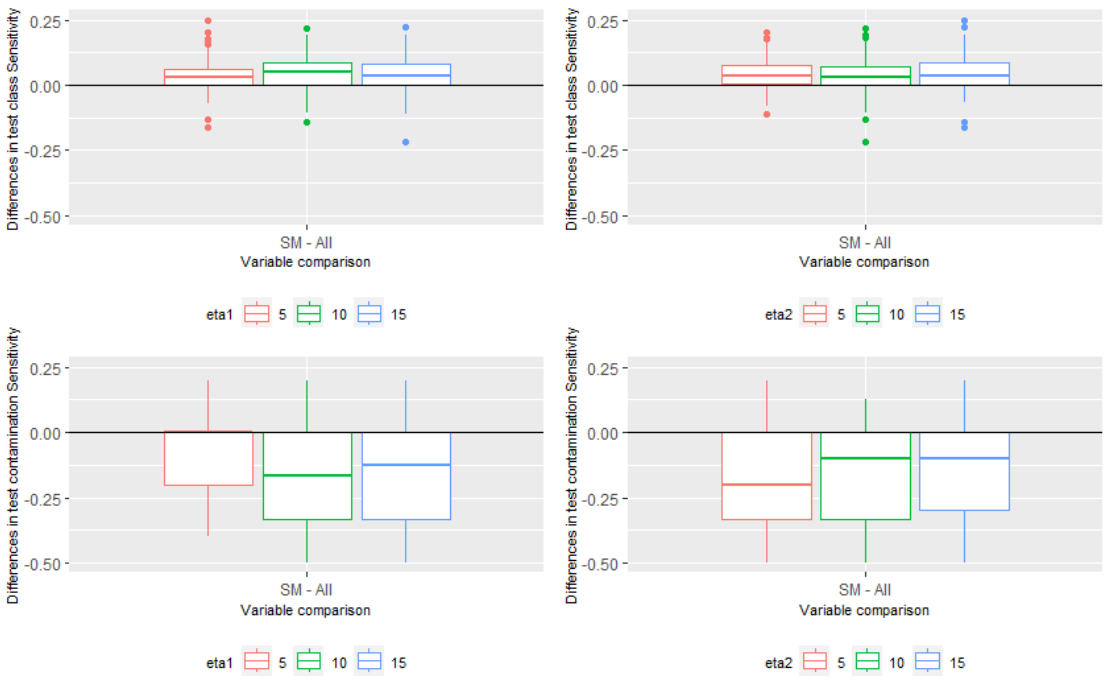



Figure 4.22: Differences in sensitivity by models on the test set at each level of $\eta$.

In Figure 4.23 the differences in specificity are positive and almost without change when $\eta_1$ and $\eta_2$ were varying at their respective levels. A few outliers were observed for the first class and for the second class a few negative outliers, which suggests that in very rare cases the "selected variables" subset underperforms compared to the "all variables" subset. Looking at the second row and assessing the contamination performance, it shows a slight increase through all differences at levels of $\eta_1$ specifically in the first class at level 15. It is also noticeable that there are a few positive extreme values at levels 5 and 15. For the second class, it is clear that differences are positive, but it appears the number of extreme values increases with an increase in $\eta_2$. Additionally, when $\eta_2$ increases the average differences in specificity between both sets of variables seems to diminish. These results suggest that the effect of varying $\eta$ is to produce occasional scenarios where the model produces poor predictions for both subsets of variables but mainly for the subset of "all variables".

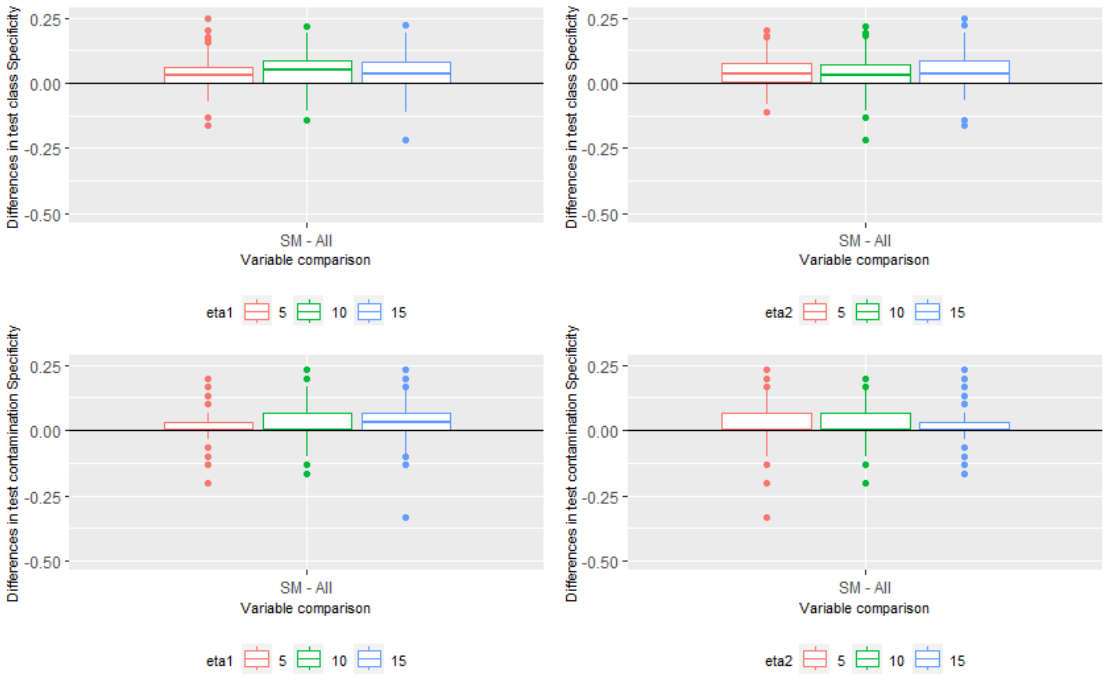



Figure 4.23: Differences in specificity by models on the test crab set at each level of $\eta$.

## 4.6 Results for the Wisconsin dataset breast cancer

The Wisconsin dataset breast cancer that was introduced in a previous chapter is a well-known used for the classification tasks. It contains features computed from digitized images of fine needle aspirates (FNA) of breast masses. These features describe characteristics of cell nuclei present in the images, such as radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension.

The dataset contains a total of 30 features, which are computed from the images. The target variable is binary and represents the diagnosis of breast cancer: malignant (cancerous) or benign (non-cancerous). This makes it a binary classification problem, where the goal is to classify whether a breast mass is cancerous or not based on the provided features.

One of the interesting features of this dataset is that it has a greater number of variables to test our proposed method. The contaminated samples were generated by calculating the sample mean and variance for both diagnoses and the proportion of each class. Then these parameters were used to generate samples from a multivariate Gaussian distribution inflating the variance-covariance matrix by allowing $\eta$ to take values $5, 10$, and 15 and

controlling the number of non-contaminated observations by allowing $\alpha$ to take values $0.75, 0.80$, and $0.85$. The results of these simulations are presented in the following section.

### 4.6.1 Overall results for all values of $\alpha$ and $\eta$

The comparison between a set of variables based on the average of the Correct Classification Rate (CCR), sensitivity, and specificity for both training and test subsets is presented in Tables 4.10, 4.11, and 4.12. The model was evaluated on the training and test subsets for the "selected variables" and "all variables".

Table 4.10 compares the average correct classification rate for the "selected" and "all variables". For the selected set of variables, the average CCR in the training set is 0.91 for the classification performance and 0.90 for contamination performance. In the testing set, these values increase to 0.95 for classification performance and 0.91 for contamination performance.

In contrast, for the 'all variables" subset, the average correct classification rate in the training set is lower, with values of 0.85 for classification and 0.87 for contamination performance. These lower values persist in the testing set, where the average correct classification rate is 0.88 for the classification performance and notably lower at 0.51 for contamination.

These results suggest that the selected set of variables generally performs better in terms of correct classification rate compared to the "all variables" set in both the training and testing phases. This highlights the importance of variable selection in improving model performance and generalization to unseen data.

Table 4.10: Comparison between a set of variables in average correct classification rate for the breast cancer data

| Metrics | Set of Variables | Train | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| CCR | Selected | 0.91 | 0.90 | 0.95 | 0.91 |
| | All | 0.85 | 0.87 | 0.86 | 0.51 |

In Figure 4.24 looking at the differences in correct classification rate between the subsets "selected variables" and 'all variables" it is noticeable that the differences are smaller in classification performance, but the differences in contamination performance

are on average just below 0.50 in favour of the "selected variables". Moreover, the range of the differences in the contamination performance shows a huge increase when it is compared with the range of differences in the classification performance.

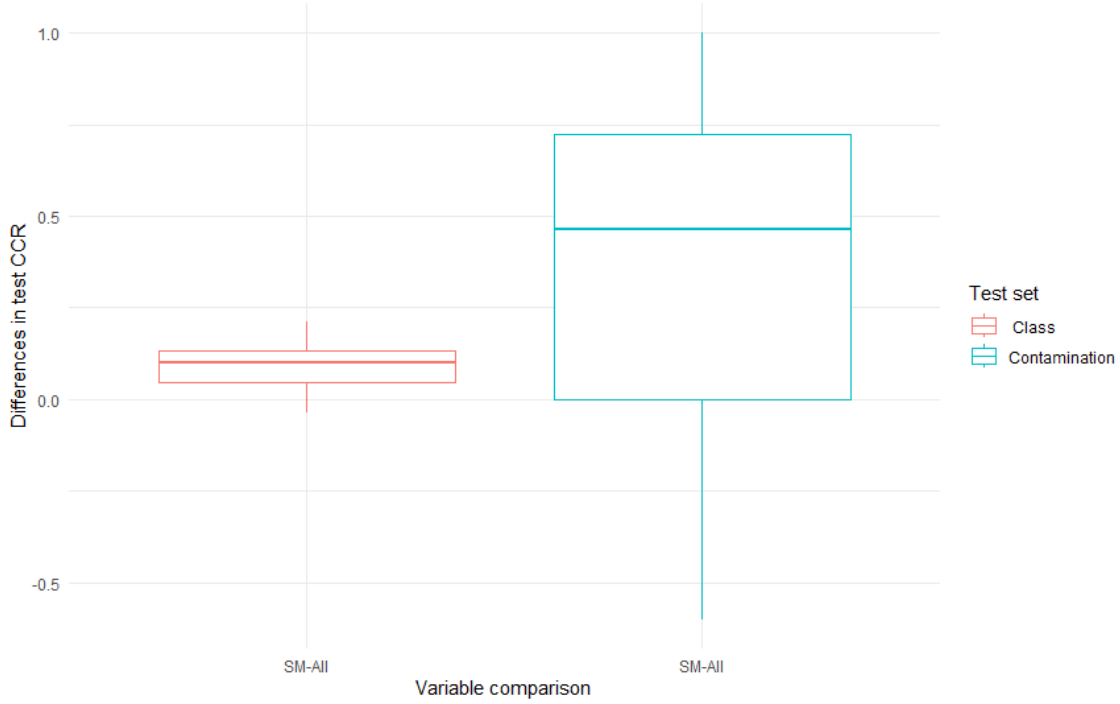



Figure 4.24: Differences in CCR by models on the test set for the Wisconsin breast cancer data.

Table 4.11 presents a comparison of average sensitivity between two subsets of variables, namely "selected variables" and "all variables" in the training and test phases for the breast cancer data.

In the training phase, the sensitivity achieved with the "selected variables" is notably higher than that with the "all variables", with values of 0.90 and 0.86 respectively, for Class contamination, indicating that the "selected variables" subset performs better in correctly identifying true positive cases. Similarly in contamination performance, the sensitivity with the "selected variables" (0.67) is higher compared to that with the "all variables" subset (0.54), suggesting that the "selected variables" subset captures positive cases while minimising false negatives.

In the test phase, the superiority of the "selected variables" subset over the "all variables" subset in terms of sensitivity is maintained. The "selected variables" exhibit higher

sensitivity values for both class and contamination performance (0.94 and 0.74, respectively) compared to the "all variables" subset (0.86 and 0.57, respectively). This indicates that the "selected variables" subset continues to outperform the "all variables" subset in correctly identifying positive cases while maintaining a lower rate of false negatives, thus demonstrating its effectiveness in capturing relevant information for classification in the breast cancer dataset.

Table 4.11: Comparison between sets of variables in average sensitivity for the breast cancer data

| Metrics | Set of Variables | Train | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| Sensitivity | Selected | 0.90 | 0.67 | 0.94 | 0.74 |
| | All | 0.86 | 0.54 | 0.86 | 0.57 |

Looking at the differences in sensitivity of the two subsets in Figure 4.25, it is possible to see that there is an increase in the range of the differences in contamination performance compared with the differences observed in the classification performance. In classification performance, although differences are small, they are positive which means that using the "selected variables" outperforms the approach of using "all variables" in classifying new observations. There were two extreme values where the differences in sensitivity between the two subsets of variables were higher than 0.2. Looking in more detail at this particular case, it was found that it might be related to the inflation factor parameters of $\eta_1$ and $\eta_2$ that were $(15, 10)$ in the first case, and $(15, 15)$ in the second case.

In contamination performance, the differences between both subsets the "selected variables" and "all variables" were positive in most cases with a small proportion of negative differences. These could be attributed to the use of 'all variables" (30 variables) possibly leading to overfitting, producing lower sensitivity.

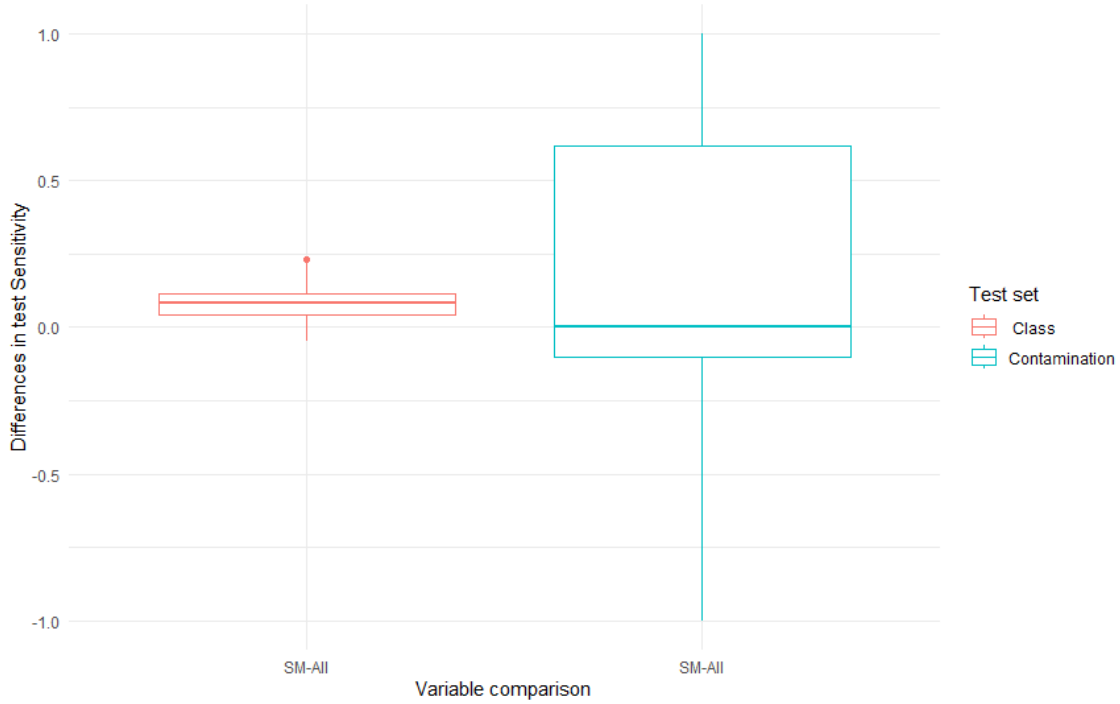



Figure 4.25: Differences in sensitivity by models on the test set for the Wisconsin breast cancer data.

Table 4.12 presents a comparison of average specificity between two subsets of variables "selected variables" and "all variables", in both the training and the test phases for breast cancer data. In the training phase, the specificity achieved with the "selected variables" subset is higher than that with "all variables" subset for both classification and contamination performance. In classification performance, the specificity values are 0.90 and 0.86 for the "selected variables" and "all variables", respectively, indicating that the "selected variables" subset performs better in correctly identifying true negative cases. Similarly, for contamination, the specificity with the "selected variables" (0.97) is higher compared to that with the "all set" variables (0.96), suggesting that the "selected variables" captures better negatives cases while minimizing false positives.

In the test phase, the superiority of the "selected variables" subset over the "all variables" subset in terms of specificity is maintained. The "selected variables" subset exhibits higher specificity values for both class and classification performances (0.94 and 0.95, respectively) compared to the "all variables" subset (0.86 and 0.90, respectively). This indicates that the "selected variables" subset continues to outperform the "all variables"

subset in correctly identifying true negative cases while maintaining a lower rate of false positives, thus demonstrating its effectiveness in capturing relevant information for classification and contamination in the breast cancer dataset.

Table 4.12: Comparison between sets of variables in average specificity for the breast cancer data

| Metrics | Set of Variables | Train | | Test | |
|---|---|---|---|---|---|
| | | Class | Contamination | Class | Contamination |
| Specificity | Selected | 0.90 | 0.97 | 0.94 | 0.95 |
| | All | 0.86 | 0.96 | 0.86 | 0.90 |

The differences in specificity of the two subsets of variables "selected variables" and "all variables" for class and contamination performance are presented in Figure 4.26. The differences in specificity between the two subsets of variables are positive for both class and contamination performance on the test set, however, they are bigger in the classification performance. For classification performance, the differences are on average just below 0.1 with two outliers that occur when $\alpha_1 = 0.8, \alpha_2 = 0.75, \eta_1 = 15$, and $\eta_2 = 10$ and 15 for the first and second outlier. The characteristic in common between these two outliers is having an inflation factor higher or equal to 10.

For contamination performance, most differences are positive and smaller than differences observed in the classification performances. Indeed, most of the differences lie below 0.1. Additionally, there are some extreme values more frequently positive with specificity values higher than 0.18 and occasionally a few negative extreme values smaller than $-0.1$. The observed negative extreme values have as common characteristics 10 and 15 as inflation for both classes. Similarly, the positive extreme values have as common characteristics percentages of non-contaminated samples $\alpha_1 = 0.75, \alpha_2 = 0.85$, and having scenarios with one of the classes taking the lowest level 5 of the inflation rate and the second class the highest value 15 and vice-versa. These results, along with what was found for sensitivity in this section, suggest a potential effect when either $\eta_1$ or $\eta_2$ is varied. Moreover, these positive differences indicate that the "selected variables" subset better captures the negative cases than the "all variables" subset.

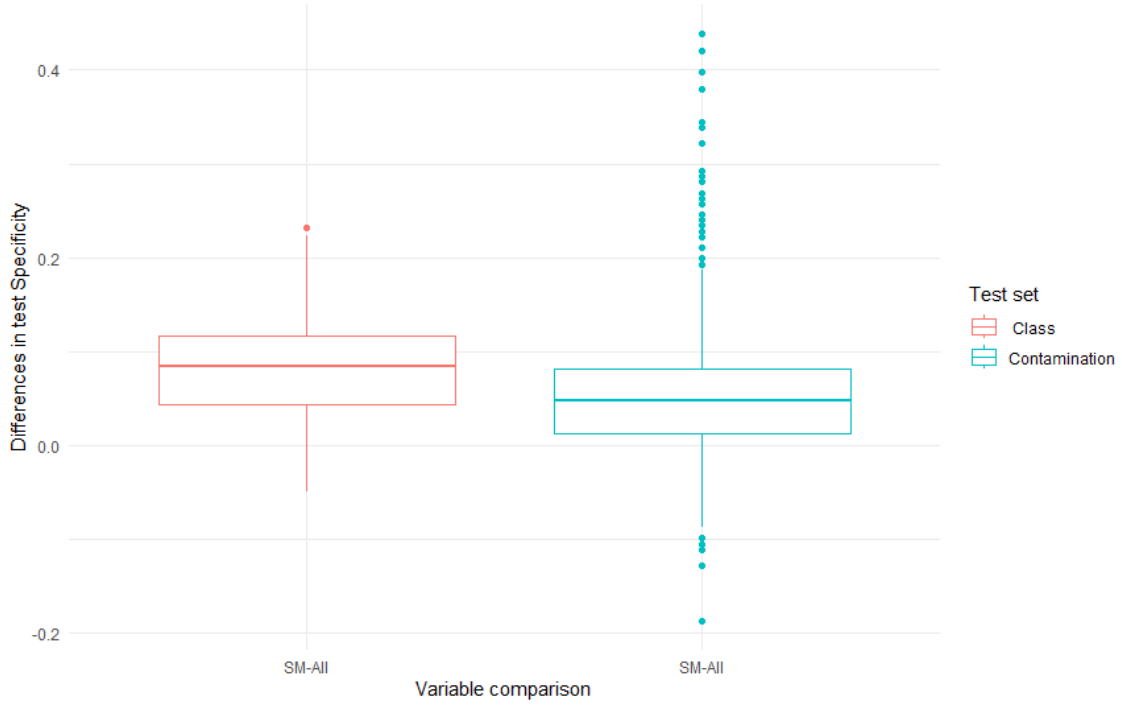



Figure 4.26: Differences in specificity by models on the test set for the Wisconsin breast cancer data.

### 4.6.2 Results for differing versus some $\alpha$

The percentage of non-contaminated observations in each sex was varied at levels $75, 80\%$, and $85\%$ to assess if the performance of the models was affected. The differences in correct classification rate, sensitivity, and specificity were recorded for both subsets of variables "selected variables" and "all variables" in classification and contamination performance.

The differences in correct classification rate for "selected variables" and "all variables" in classification and contamination performance at different levels of $\alpha_1$ and $\alpha_2$ are shown in Figure 4.27. In the first row of the plot containing the CCR for classification performance, it can be seen that all the differences are positive for both classes of benign and malignant tumors. However, there seems to be a pattern in each class. When $\alpha_1$ increases, the positive differences also increase. Contrary, in the second class when $\alpha_2$ increases, the differences decrease. Moreover, some outliers are visible and the plot suggests a potential effect on the differences when either $\alpha_1$ or $\alpha_2$ take values 0.75 and 0.85. Additionally, these outliers have in common having at least one of the classes an inflation factor of 15

Looking at contamination performance in the second row of the plot, there is not a clear pattern. It seems that for the first class setting $\alpha_1$ at values 0.75 or 0.85 produces some cases with positive differences. For the second class, it seems that setting $\alpha_2$ either to 0.75 or 0.80 produces positive differences while setting $\alpha_2$ to 0.85 produces negative differences. These results illustrate the better performance of the "selected variables" in classification for both classes. In contamination the "selected variables" performs better for the first class at levels of $\alpha_1 = 0.75, 0.85$ while in the second class at levels $\alpha_2 = 0.75, 0.85$. The subset of "all variables" performs better than the subset "selected variables" in the first class when $\alpha_1 = 0.80$ while for the second class at level $\alpha_2 = 0.85$. Also, there are occasional outliers for the first class at levels $\alpha_1 = 0.75, 0.85$ and for the second class at level $\alpha_2 = 0.85$.

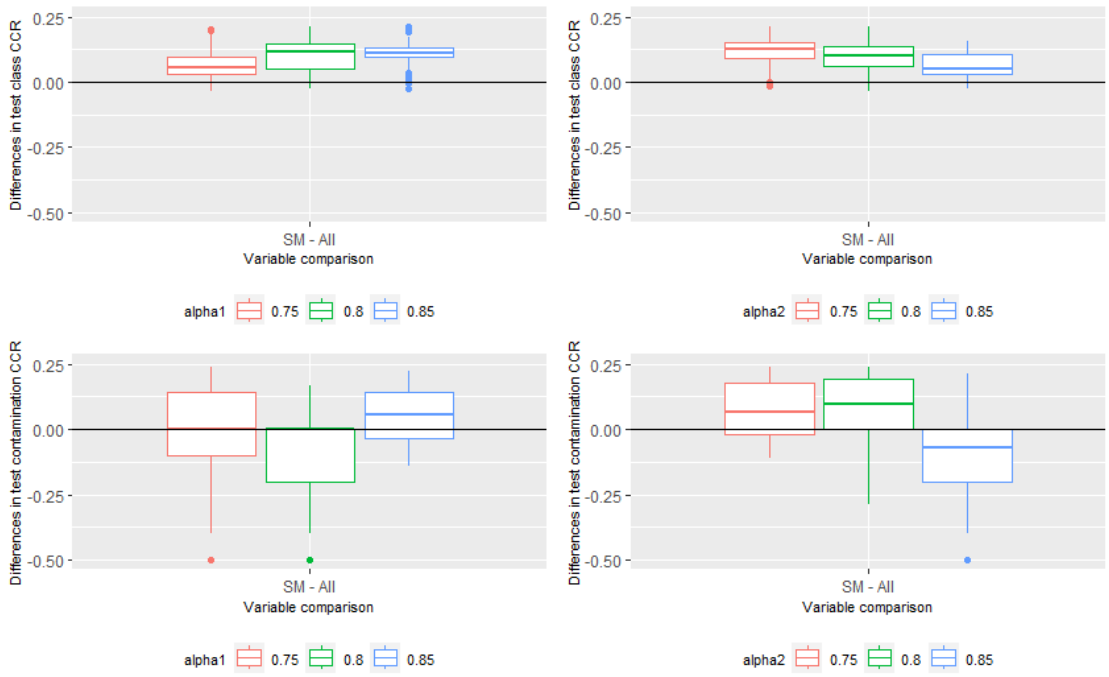



Figure 4.27: Differences in CCR by models on the test set at each level of $\alpha$ for the Wisconsin breast cancer data.

Figure 4.28 shows the differences in sensitivity for the "selected variables" and "all variables" in classification and contamination performance at different levels of $\alpha_1$ and $\alpha_2$.

Looking at the first row in Figure 4.28, it is clear that in the classification performance,

all differences are positive. The plots in the first row display an increase in sensitivity for the first class. Contrary to the first class, the second class displays a decrease in sensitivity with an increase in the percentage of non-contaminated observations.

In the contamination performance (second row), all differences are negative. For the first class an increase of $\alpha_1$ reduces the differences. Additionally, increasing $\alpha_1$ to 0.85 appears to produce scenarios with outliers in the differences. For the second class an increase of $\alpha_2$ increases the differences. Moreover, it appears that setting $\alpha_2 = 0.75$ shows some extreme values. The results reflect the better classification performance of the subset of "selected variables" than the subset of "all variables". Contrary to their better performance of the subset of "selected variables", the subset of "all variables" yields higher sensitivity in contamination performance.

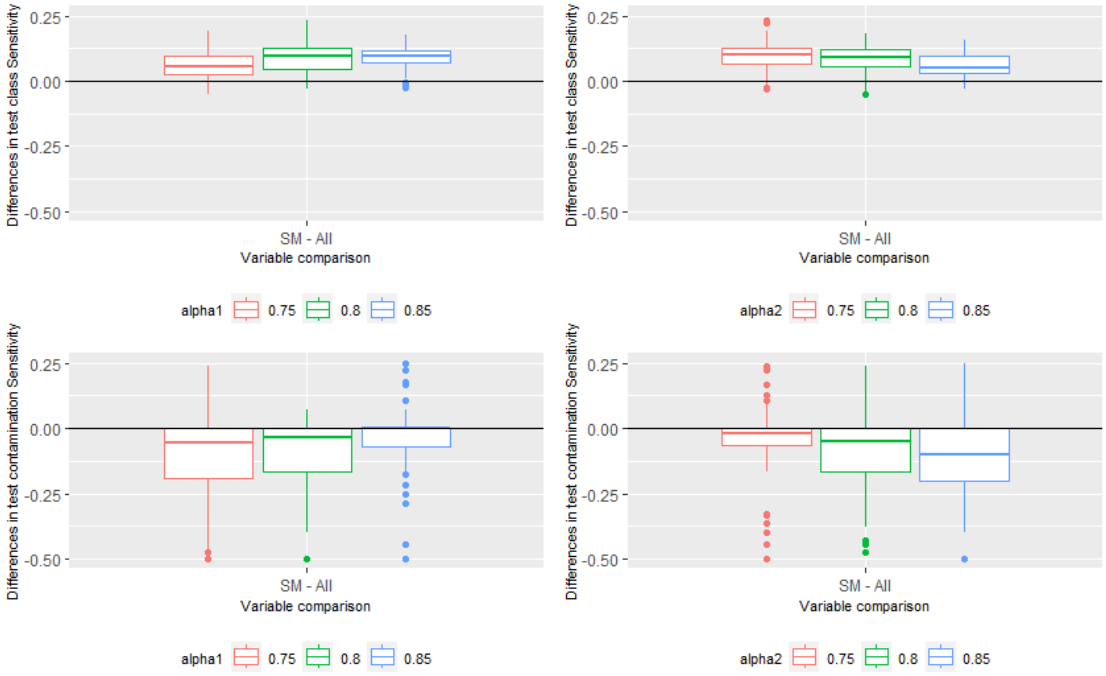



Figure 4.28: Differences in sensitivity by models on the test set at each level of $\alpha$ for the Wisconsin breast cancer data.

Figure 4.29 illustrates the differences in specificity for the "selected variables" and "all variables" in classification and contamination performance at different levels of $\alpha_1$ and $\alpha_2$.

Looking at the classification performance (first row) in Figure 4.29, the results illustrate

positive differences in specificity for classification performance for all classes regardless of the level of $\alpha_1, \alpha_2$. Additionally, in the first class, there is an increase of differences in specificity when $\alpha_1$ increases. Contrary to the first class, a decrease of differences in specificity is observed as $\alpha_2$ increases. In contamination performance, for the first class, it appears that an increase in $\alpha_1$ produces an increase in differences in specificity. For the second class, it appears that an increase in $\alpha_2$ yields a slight decrease in differences in specificity. Moreover, some outliers were observed which have as common factor values of $\eta_1$ equal to 10 or 15.

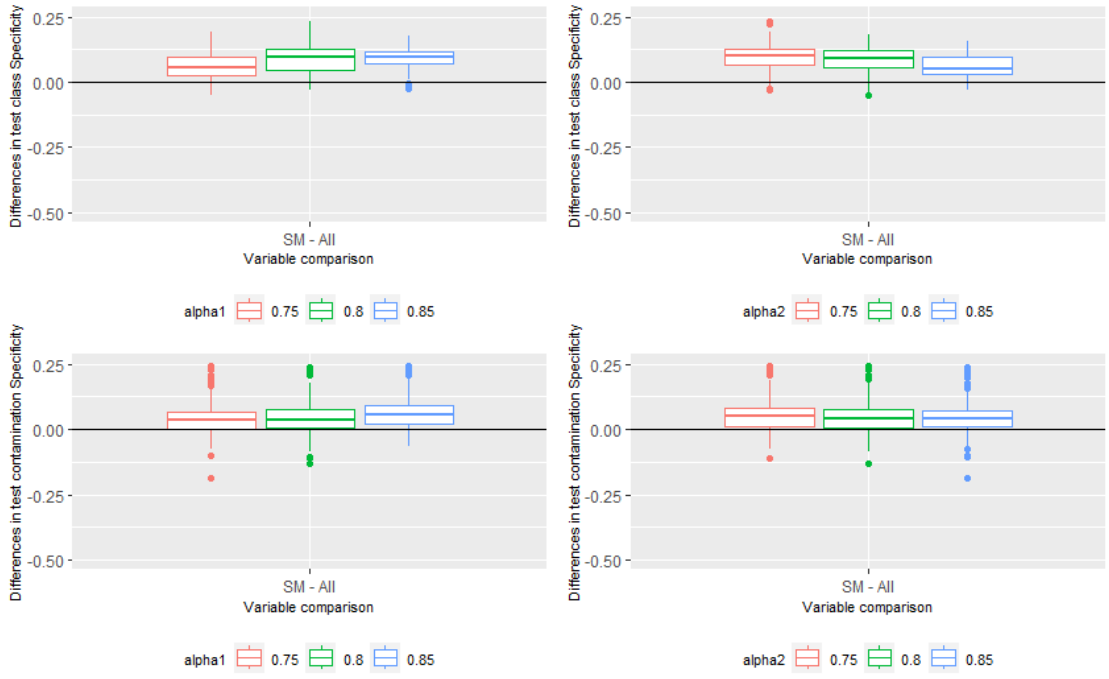



Figure 4.29: Differences in specificity by models on the test set at each level of $\alpha$ for the Wisconsin breast cancer data.

### 4.6.3 Results for differing versus some $\eta$

The inflation factor $\eta$ was varied at levels $5, 10$, and 15 to assess if the performance of the variable selection is affected. The differences in performance metrics of two sets of variables are examined for classification and contamination performance on the breast cancer dataset

In Figure 4.30 the differences in correct classification rate for the "selected variables" and "all variables" in classification and contamination performance at different levels of

$\eta_1$ and $\eta_2$ are displayed. For the classification performance (first row), the differences are positive which means a better performance of the "selected variables" subset over the "all variables" subset. It appears that there is a slight increase in CCR with an increase in $\eta_1$ and a slight decrease with an increase in $\eta_2$.

In contamination performance (second row), the results suggest a decrease in differences in CCR with an increase in $eta_1$ and an increase in CCR when $\eta_2$ increases.

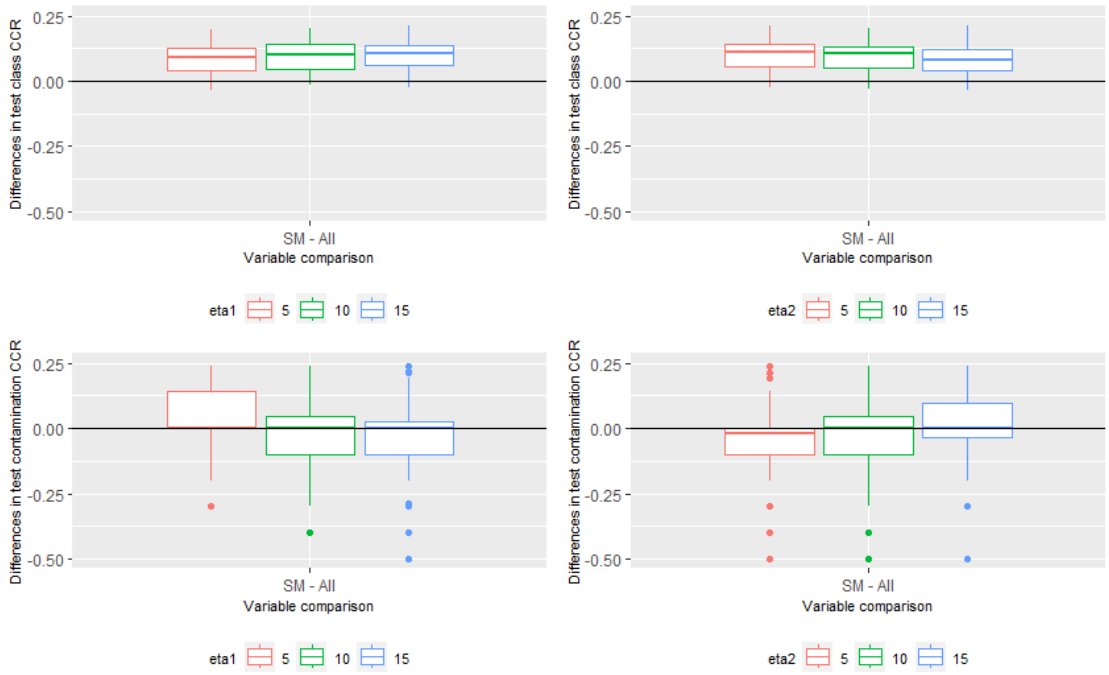



Figure 4.30: Differences in CCR by models on the test set at each level of $\eta$ for the Wisconsin breast cancer data.

The differences in sensitivity between the subset "selected variables" and "all variables" for classification and contamination performance are presented in Figure 4.31. In the classification performance (first row), it can be seen that all the differences are positive for both classes suggesting a better performance of the subset of the "selected variables" capturing true positive than the subset of "all variables". In the contamination performance (second row), differences for both classes are negative. A decrease is observed in the differences with an increase in $\eta_1$ and an increase in differences with an increase in $\eta_2$. These results suggest a better performance of the subset of "all variables" capturing true positives in identifying contaminated observations. Additionally, the presence of outliers

is observed, especially at $\eta = 5$ for the first class and at $\eta_2 = 15$ for the second class.

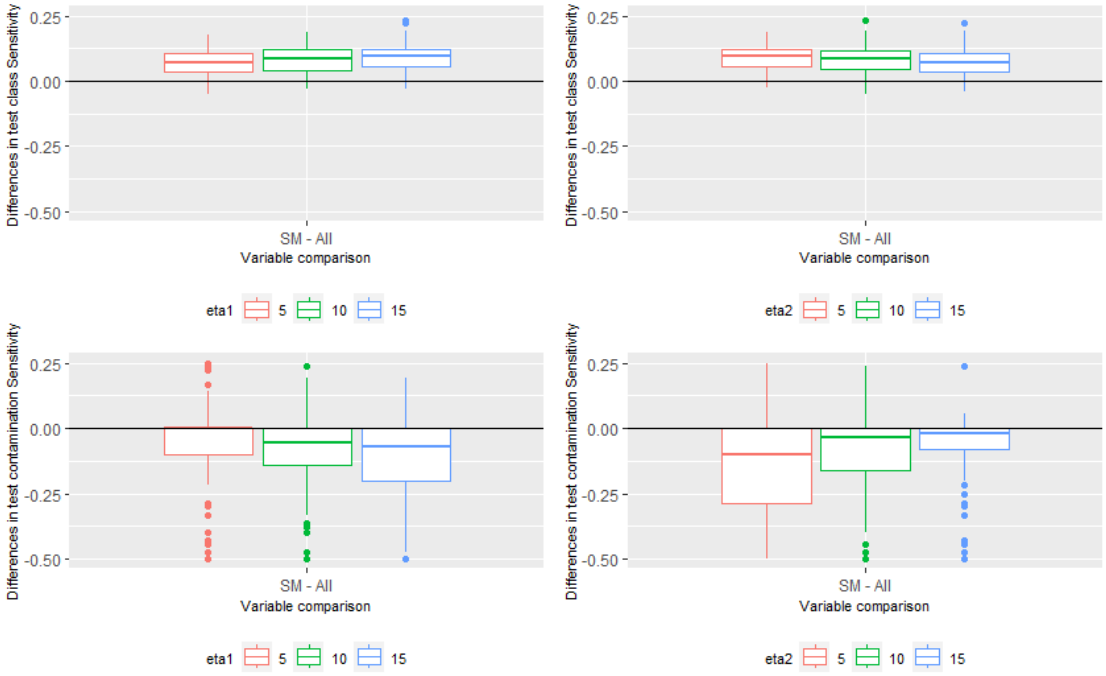



Figure 4.31: Differences in sensitivity by models on the test set at each level of $\eta$ for the Wisconsin breast cancer data.

The differences in specificity between the subset "selected variables" and "all variables" for the classification and contamination performance are presented in Figure 4.32. In the classification performance (first row), it can be seen that all the differences are positive for both classes suggesting a better performance of the subset of the "selected variables" capturing true positive than the subset of "all variables". In the contamination performance (second row), the differences are also positive for both classes regardless of values $\eta_1$ and $\eta_2$. Additionally, some extreme values are observed in the contamination scenario regardless of the value of $\eta$.

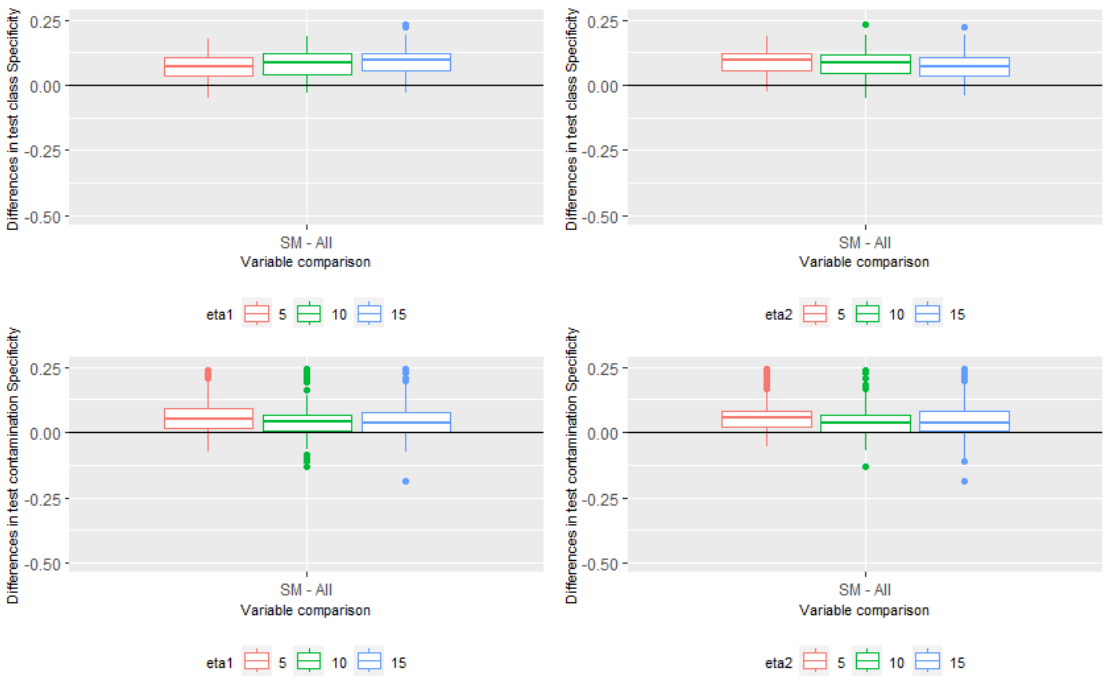



Figure 4.32: Differences in specificity by models on the test set at each level of $\eta$ for the Wisconsin breast cancer data.

## 4.7 Discussion

The semi-supervised mixture of Gaussian models has seen increased attention in the last few years (Yan, 2017; Śmieja, 2020) and its application has expanded to many different fields from speech recognition to image segmentation. However, there has been some resistance to exploring semi-supervised methods due to the criticism they face regarding their dependence on unlabeled data. Another possible reason for less work on semi-supervised learning is reports that having a vast amount of unlabeled data does not always help improve model performance. Understanding in which cases unlabeled data contributes to improvement and in which cases it does not has raised interest. Recently there has been some light on the answer to this question with the work done by Castelli and Cover (1995), Singh et al. (2008), Yang and Priebe (2011). This work along with others reflects a growing interest in semi-supervised methods.

The objective of this chapter was to expand the variable selection method proposed for supervised learning in Chapter 3 to its corresponding one in the semi-supervised learning case. A mixture of supervised contaminated Gaussian distributions was chosen as a refer-

ence model and half of the observations in the training data are considered unlabeled data.

To facilitate the comparison with the variable selection for a supervised mixture of contaminated Gaussian distributions, the model was applied to the same data to which its supervised version was exposed in Chapter 3. The results were quite similar, confirming that the selection of variables in the worst case does not affect the test class correct classification rate, although it was evident, as in Chapter 3, that there is a deterioration in the test contamination sensitivity which is the ability to recognize contaminated observations. The main reason for this case is that when contamination is present in all variables, the variable selection procedure discards variables that are irrelevant for classification but might be relevant to identifying contamination in the samples.

Analyzing the main factors that affect the performance of the model, it was found that the main positive factor to improve the correct classification rate indicator was the presence of unbalanced classes. This occurs because the model has more training observations of the predominant class, so the CCR improves in this class to the detriment of the smaller class. The second factor that had a positive effect was using variable selection followed by having an additional variable that separates the classes.

Among the factors that had the most negative impacts on the correct classification rate were having classes whose means are at a very close distance. This agrees with what was reported by Steinley and Brusco (2008). The other factor that negatively impacted CCR was a high correlation between two variables that created the separation between classes, which has also been reported as deteriorating performance of models by Derkseen and Kesselman (265–282).

There were simulations where some extreme values were observed in the boxplots showing the differences in the metrics corresponding to each variable set. The main common factor behind these scenarios was unbalanced classes that led to training sets with a small number of contaminated observations for training in the less represented class. This combined with other factor levels that negatively affect performance metrics such as a strong correlation between variables that create class separation, or classes whose means are very close, produced differences in the performance obtained by each subset of variables ana-

lyzed.

The plasmode datasets were used to test the model with data that contain some of the challenges of real problems and since there was no confirmed presence of contaminated observations, contaminated observations were simulated that were later included as part of the data set to be analysed. The closest scenario to a typical semi-supervised learning scenario was the crab species dataset. This small dataset is made up of just 100 observations, 50 males and females, which when divided between 75% for training and 25% for test. In this dataset, half of the observations in the data set were considered unlabeled observations. The results showed that when comparing the performance of the model in training and in the test subsets, there was an increase of 3 percentage points in sensitivity for class prediction and 12 percentage points of sensitivity in the identification of contaminated observations.

Wrapper methods are criticized for being considered brute force, but also for being computationally demanding. This was partly a limitation that needed managing. For example rather than looking at all possible covariance structures, the contaminated Gaussian mixture model was only allowed to choose between the following covariance structures: EEI, VII, VEI, EEI, EEE, VVV. The CPU time it took for the variable selection method to run on a data set composed of 3000 observations in 3 classes mapped in 100 dimensions, allowing choice between the covariance models EEI, VII, VEI, EEI, EEE, VVV, using a laptop with 15 GB of RAM and a 4.4 GHz processor was around 14 hours.

The greedy search algorithm was able to identify and select the variables that created the separation between classes in the simulated datasets. It also discarded the majority of non-informative classification variables. Most of the selected subsets were composed of less than 7 variables.

The results obtained via simulation studies show that if there is evidence of the same level of contamination across all observed variables and it is feasible to build a model including all the variables, a model formed by all the variables will give the best course of action when the object of interest is identifying contamination not classification performance. However, if it is not possible to include all the available variables, the advice will be to use a semi-supervised mixture of Gaussian models wrapped in a variable selection

algorithm.

The proposed forward greedy search algorithm can be tailored to extend mixtures of Laplace distributions, mixtures of t-distributions (Andrews and McNicholas, 2012), mixtures of skewed Gaussian distributions, and mixtures of skewed t distributions to deal with high-dimensional data. These alternatives mixtures are appropriate in scenarios where the contaminated component of each group contains observations with longer than Gaussian tails. The assumption that each class follows a particular distribution might not be adequate in some scenarios and might the mixture might fail to capture the data structure. Therefore, it would be appealing to explore an alternative that does not make this assumption.

Although it might be possible to explore the extension of the proposed method to cases where the data exhibits strong non-linear relationship through an artificial intelligence method (Jiang et al., 2016; Xie et al., 2016), there are some challenges to consider such as: the scarcity of training samples, the unknown contaminated labels, the lack of identification of the features that create the separation between classes, and the execution time that might require a complex neural network.

# Chapter 5

# Conclusions

In real-life applications, it is not always possible to assume the absence of outliers in the data. Hence, assuming that the data is contaminated becomes relevant since outliers can deteriorate the performance of classification models by affecting the estimation of their parameters (Barnett et al., 1994; Gallegos and Ritter, 2009). There are different ways of treating outliers in data but there are certain types of classification problems, especially in food authenticity where the particular interest is to model and identify them instead of discard them.

The idea of the contaminated Gaussian distribution, having a two-component mixture of Gaussian distribution, with one component modelling "good observations" and the other offering protection against mild outliers is not new and it can be traced back to the work done by Tukey (1960), Aitkin and Wilson (1980), and lately by the introduction of a mixture of contaminated Gaussians by Punzo and McNicholas (2016). This type of model deals specifically with cases where the contamination can be treated as mild outlying observations (Ritter, 2014) that differ only a little from the reference model. In cases of extreme outliers, this modelling approach is not recommended.

This thesis focused on extending the application of a mixture of contaminated Gaussian distributions to scenarios with a large number of variables or a great portion of unlabelled data. The methodological work of this thesis can be broken into two parts:

- the extension of a multivariate of contaminated Gaussian distributions to fit datasets with a large number of variables via variable selection in a supervised setting.
- variable selection for a semi-supervised mixture of Gaussian distributions.

Chapter 2 gives an overview of relevant work dealing with contaminated data and intro-

duces some of the challenges. Two main challenges are presented: 1) the need to overcome overfitting the model, and 2) the need for a procedure that selects the most informative variables while discarding the non-informative ones. The work related to the extension of the contaminated mixture of Gaussian distributions to deal with high-dimensional data is presented in Chapter 3, and is summarised here in Section 5.1. The semi-supervised mixture of contaminated Gaussian distribution was proposed and evaluated in Chapter 4, and is summarised here in Section 5.2,

## 5.1 Variable selection for a supervised mixture of contaminated Gaussian distributions

The mixture of contaminated Gaussians model can in practice often be limited to datasets with a small number of variables due to the large number of parameters. The choice can sometimes come down to having to fit a highly restricted, simplified, and unrealistic model due to high-dimensional data or not being able to use the model at all.

To overcome that challenge a wrapper was tailored in the supervised setting to a mixture of contaminated Gaussian distributions in Section 2.9.2. The direction of search in this thesis' approach was a forward one as this made sense in terms of starting with smaller models rather than a backward search which would start with a model with all variables included, which may not be possible to fit depending on the number of variables, and removing variables in turn. However, in cases with medium number of variables, a stepwise approach which alternates between inclusion and exclusion steps might be possible. Additionally, instead of a greedy search method, an alternative search method such as headlong search (Cormen T, 2022) could also be considered, whereby rather than choosing the best variable for inclusion at each stage of the search, variables are checked in turn until the first variable found to improve the model is selected. In order for this to work well, some sort of ordering of the variables in terms of potential classification performance needs to be done, as the order in which the variables are examined will matter in the headlong search where it did not in the greedy search. All of these search methods, however, are local search methods which potentially can miss the global optimum in favour of a local optimum.

A simulation framework was established to assess the proposed model in different scenarios created by varying factors that have been identified as influential in the performance of classification models (Steinley and Brusco, 2008). The results of these simulations show in general that the use of variable selection does not deteriorate the class correct classification rate, but there is a price to pay in sensitivity for extending the mixture of contaminated Gaussian distributions to high-dimensional data since using variable selection yields lower levels of sensitivity when it was compared to using the set of "all variables". To understand the effect of influential factors on the CCR a model was fitted with the CCR as response variable and factors as predictors. The regression coefficients of the model revealed that assuming the best scenario where the variables that create the separation between classes are known, including the variables obtained by a variable selection, will improve the CCR positively by 0.02 units. Additionally, adding a variable search step improved the test contamination sensitivity by 0.54. Moreover, the simulation results in Section 3.3 show that the greedy search algorithm produces sensible subsets the vast majority time, which include the variables that create separation between classes and exclude the majority of irrelevant variables.

Additionally, the results of the simulation and plasmode studies conducted in Sections 3.3, 3.5, 3.6, and 3.7 support the conclusion that the selected variables yielded a higher class correct classification rate in some of the scenarios considered and that there is a loss in test contamination sensitivity that comes with performing variable selection. Although in many scenarios it is still possible to identify a decent percentage of the contaminated observations.

## 5.2 Variable selection for a semi-supervised mixture of contaminated Gaussian distributions

In many real data cases, there are much fewer labelled observations (data with known group information) compared to the amount of unlabelled data. Classification approaches typically only use labelled data to create the model used to predict group membership (or contamination) but this ignores the information in the unlabelled data. Semi-supervised learning looks to incorporate unlabelled data within the classification model parameter estimation. Although semi-supervised methods have been criticised for their reliance on

unlabelled data, it is worth noting that some of the approaches that have shown great success in challenging artificial intelligence (AI) tasks, as demonstrated by Bengio et al. (2013); Hinton and Salakhutdinov (2006), which are constructed upon unsupervised learning algorithms, and Erhan et al. (2010).

The simulation study results support the inclusion of variable selection for most of the scenarios with a contribution of 0.01 units on CCR. However, there is a concern about applying a variable selection for supervised/semi-supervised mixtures of contaminated Gaussian distributions to datasets with unbalanced classes. It is suggested to treat the unbalanced classes issue before applying the proposed methods. In the rest of the simulated cases, the proposed method was able to produce similar performance to using the "all variables" subset in classification. In terms of prediction contamination, there was a loss of sensitivity due to the exclusion of some variables that did no contain class information but they were relevant for contamination.

When the proposed variable selection approach was tested with the plasmode datasets, the results were consistent with the findings obtained in the study simulations. Each of the plasmode datasets offers different challenges, for example, the crab data is a dataset with a small number of variables but high correlation, and variable selection with a mixture of contaminated Gaussians performs well in predicting class membership and identifying contaminated samples. The results obtained in the Wisconsin breast cancer dataset also suggest a good performance in correct classification rate in the training and the test sets, however, there is a decrease in class contamination sensitivity. Overall, using the selected variables does not underperform compared to incorporating all the variables in terms of test class correct classification rate and test contamination correct classification rate. However, there was a noticeable effect of the percentage of non-contaminated samples along with the inflation factor on the CCR, since they control the number of contamination observations available for training and test and also the dispersion of the contaminated samples with respect to the non-contaminated ones.

## 5.3 Limitations and future work

The simulated data was generated from a mixture of contaminated Gaussian distributions and met the normality assumptions that the proposed model required. Hence, the proposed method has some limitations such as it could break down with extreme outliers (Hennig,

2004). There are different approaches for this, such as screening for the values that will break the model to exclude them from analysis or adopting a trimmed likelihood estimator (Neykov et al., 2007) or a similar method that down-weights the contribution of these observations in the parameter estimation. Moreover, a standardization of the data will be a good practice along with excluding variables that do not have much variability to correct little deviation from normality in datasets.

An obvious extension to speed up the variable selection procedure to manage even a much larger set of variables will be using another search strategy or discarding a good number of irrelevant variables such apply feature selection via discretization (Liu and Setiono, 1997) or using variable parameter shrinkage via regularization (Celeux et al., 2019). Additionally, nowadays with PCs having access to Graphics Processing Units (GPUs) which are reported to be much faster than Central Processing Units (CPUs), it will be interesting how much a greedy search algorithm can be sped up by coding to take advantage of this.

The variable selection proposed here comes with a trade-off against sensitivity in detecting contaminated observations. This is because the variable selection criterion only incorporates a criterion to assess how the model will perform in terms of correct classification rate. It is not attempting to optimize contamination detection. It was clear in some of the individual simulation plots also, that increasing levels of contamination often meant more overlap of classes, so the method focusing on variables that separated classes would likely remove variables that best identified contamination. Having a mixed criterion that takes into account the gains of a variable in terms of contamination sensitivity (e.g likelihood of an unsupervised mixture of contaminated Gaussians) might reduce the loss of sensitivity in identifying contaminated samples.

To sum up, even in datasets where some of their characteristics might harm the performance of the model such as the strongly correlated crab data set, the proposed model produces a good level of classification and discrimination of contaminated observations. In real scenarios the variables that create class separation are unknown, and using all available variables is not practical since it deteriorates the classification model. Therefore, the proposed method gives a valid approach to extending the mixture of contaminated Gaussian distributions to identify contaminated samples in high dimensional data.

# Appendix A

# Additional results from Chapter 4

In this appendix, additional results corresponding to Chapter 4 are presented, specifically from the regression models modelling the correct classification rate.

### A.0.1 Modelling mean test class of correct classification rate (CCR) by factor

looking at the residuals from the model in Section 4.3.8 it is clear that there is a deviation from the assumption of normality in the residuals of the model. This might not allow for inference confidence intervals for the parameters as the p-values would not be reliable, but the estimates for the coefficients are still valid.

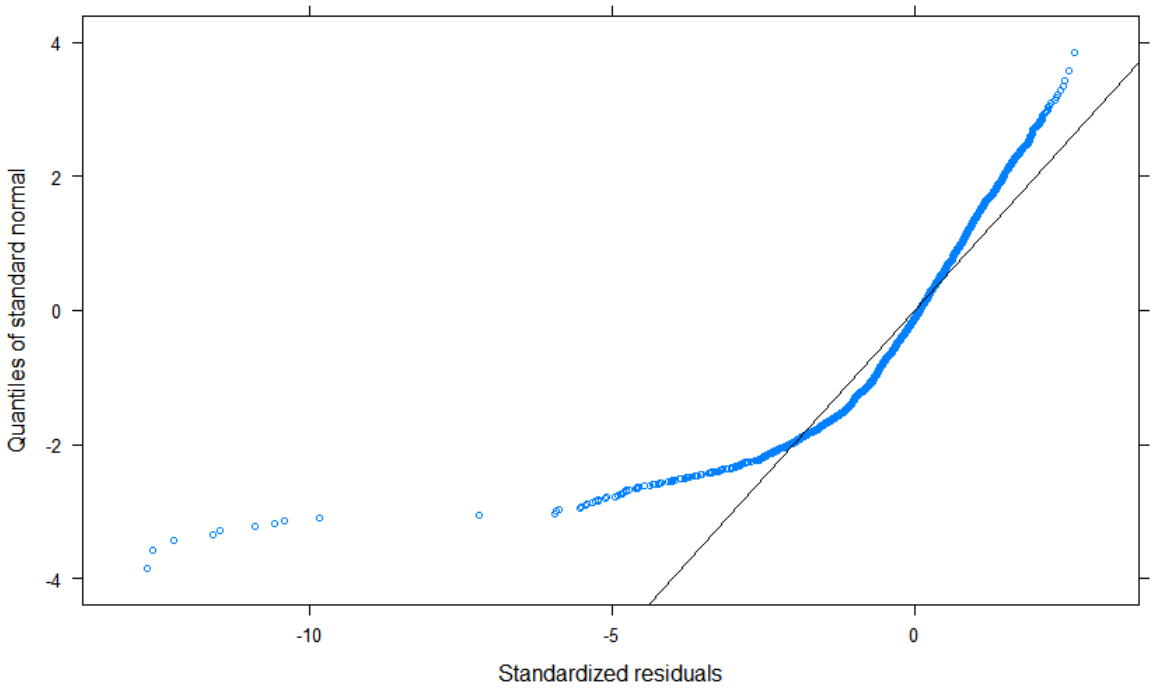



Figure A.1: Q-q plot for the model

.

# References


Adrian, R. (1996), 'Hypothesis testing and model selection via posterior simulation', *Markov chain Monte Carlo in practice* .

Aeberhard, S. and Forina, M. (1991), 'Wine', UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J.

Aitkin, M. and Wilson, G. T. (1980), 'Mixture models, outliers, and the em algorithm', *Technometrics* **22**(3), 325–331.

Akaike, H. (1974), 'A new look at the statistical model identification', *IEEE transactions on automatic control* **19**(6), 716–723.

Albaseer, A., Ciftler, B. S., Abdallah, M. and Al-Fuqaha, A. (2020), 'Exploiting unlabeled data in smart cities using federated learning', *arXiv preprint arXiv:2001.04030* .

Anderson, J. (1979), 'Multivariate logistic compounds', *Biometrika* **66**(1), 17–26.

Andrews, J. L. and McNicholas, P. D. (2012), 'Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions: the t eigen family', *Statistics and Computing* **22**, 1021–1029.

Andrews, J. L., McNicholas, P. D. and Subedi, S. (2011), 'Model-based classification via mixtures of multivariate t-distributions', *Computational Statistics & Data Analysis* **55**(1), 520–529.

Azzalini, A. (2013), *The skew-normal and related families*, Vol. 3, Cambridge University Press.

Banfield, J. D. and Raftery, A. E. (1992), 'Ice floe identification in satellite images using mathematical morphology and clustering about principal curves', *Journal of the American Statistical Association* **87**(417), 7–16.

Banfield, J. D. and Raftery, A. E. (1993), 'Model-based gaussian and non-gaussian clustering', *Biometrics* pp. 803–821.

Barnett, V., Lewis, T. et al. (1994), *Outliers in statistical data*, Vol. 3, Wiley New York.

Becker, C. and Gather, U. (1999), 'The masking breakdown point of multivariate outlier identification rules', *Journal of the American Statistical Association* **94**(447), 947–955.

Bengio, Y., Courville, A. and Vincent, P. (2013), 'Representation learning: A review and new perspectives', *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828.

Biernacki, C. and Govaert, G. (1999), 'Choosing models in model-based clustering and discriminant analysis', *Journal of Statistical Computation and Simulation* **64**(1), 49–71.

Binder, D. A. (1978), 'Bayesian cluster analysis', *Biometrika* **65**(1), 31–38.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, Springer.

Blum, A. L. and Langley, P. (1997), 'Selection of relevant features and examples in machine learning', *Artificial intelligence* **97**(1-2), 245–271.

Bock, H.-H. (2002), 'Clustering methods: From classical models to new approaches', *Statistics in Transition* **5**(5), 725–758.

Böhning, D. (2000), 'Computer-assisted analysis of mixtures and applications'.

Bouveyron C, Celeux G, M. T. R. A. (2019), *Model-based Clustering and Classification for Data Science with applications in R.*

Bouveyron, C., Girard, S. and Schmid, C. (2007), 'High-dimensional data clustering', *Computational statistics & data analysis* **52**(1), 502–519.

Bruce, R. F. (2001), A bayesian approach to semi-supervised learning., *in* 'NLPRS', pp. 57–64.

Burnham, K. P. and Anderson, D. R. (2004), 'Multimodel inference: understanding aic and bic in model selection', *Sociological methods & research* **33**(2), 261–304.

C Ruwet, L A Garcia-Escudero, A. G. A. M.-I. (2012), 'The influence function of the tclust robust clustering procedure', *Advances in Data Analysis and Classification* **6**, 107–130.

Campbell, N. (1984), 'Mixture models and atypical values', *Journal of the International Association for Mathematical Geology* **16**, 465–477.

Campbell, N. and Mahon, R. (1974), 'A multivariate study of variation in two species of rock crab of the genus leptograpsus', *Australian Journal of Zoology* **22**(3), 417–425.

Cappozzo, A., Greselin, F. and Murphy, T. B. (2021), 'Robust variable selection for model-based learning in presence of adulteration', *Computational statistics & data analysis* **158**, 107186.

Castelli, V. and Cover, T. M. (1995), 'On the exponential value of labeled samples', *Pattern Recognition Letters* **16**(1), 105–111.

Castelli, V. and Cover, T. M. (1996), 'The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter', *IEEE Transactions on information theory* **42**(6), 2102–2117.

Celeux, G. (1995), 'Gaussian parsimonious clustering models', **28**(5), 781–793.

Celeux, G., Maugis-Rabusseau, C. and Sedki, M. (2019), 'Variable selection in model-based clustering and discriminant analysis with a regularization approach', *Advances in Data Analysis and Classification* **13**, 259–278.

Chapelle, O., Scholkopf, B. and Zien, A. (2006), *Semi-supervised learning. 2006*, Vol. 2.

Cormen T, Leisersson C, R. T. S. C. (2022), *Introduction to algorithms*, MIT press.

Danezis, Tsagkaris A, C. F. B. V. G. C. (2016), 'Food authentication: Techniques, trends emerging approaches', *Trends in analytical chemistry* (85), 123–132.

Dasgupta, A. and Raftery, A. E. (1998), 'Detecting features in spatial point processes with clutter via model-based clustering', *Journal of the American statistical Association* **93**(441), 294–302.

Dean, Brendam M, D. G. (2006), 'Using unlabelled data to update classification rules with applications in food authenticity studies', *Royal Statistical Society. Series C(Applied Statistics)* **55**(1), 1–14.

Derkseen, S. and Kesselman, H. J. (265–282), 'Backward, forward, and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables', *British Journal of Mathematical and Statistical Psychology* **45**(2).

Dua, D. and Graff, C. (2019), 'UCI machine learning repository'.

Dubey, A. K., Gupta, U. and Jain, S. (2016), 'Analysis of k-means clustering approach on the breast cancer wisconsin dataset', *International journal of computer assisted radiology and surgery* **11**, 2033–2047.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000), *Pattern Classification (2nd Edition)*, 2 edn, Wiley-Interscience.

Edwards, A. W. and Cavalli-Sforza, L. L. (1965), 'A method for cluster analysis', *Biometrics* pp. 362–375.

El Bouchefry, K. and de Souza, R. S. (2020), Learning in big data: Introduction to machine learning, *in* 'Knowledge discovery in big data from astronomy and earth observation', Elsevier, pp. 225–249.

Erhan, D., Courville, A., Bengio, Y. and Vincent, P. (2010), Why does unsupervised pre-training help deep learning?, *in* 'Proceedings of the thirteenth international conference on artificial intelligence and statistics', JMLR Workshop and Conference Proceedings, pp. 201–208.

Eusebi (2013), 'Diagnosted accuracy measures', *Cerebrovascular diseases* (36), 267–272.

Fisher, R. A. (1936), 'The use of multiple measurements in taxonomic problems', *Annals of eugenics* **7**(2), 179–188.

Forina, M., Leardi, R., Armanino, C., Lanteri, S., Conti, P., Princi, P. et al. (1988), 'Parvus an extendable package of programs for data exploration, classification and correlation'.

Fraley, C. and Raftery, A. (1998*a*), 'Mclust: Software for model-based cluster and discriminant analysis', *Department of Statistics, University of Washington: Technical Report* **342**, 1312.

Fraley, C. and Raftery, A. (2002), 'Model-based clustering, discriminant analysis, and density estimation', *Journal of the American Statistical Association* **97**(458), 611–631.

Fraley, C. and Raftery, A. E. (1998*b*), 'How many clusters? which clustering method? answers via model-based cluster analysis', *The computer journal* **41**(8), 578–588.

Fraley, C. and Raftery, A. E. (2007), 'Bayesian regularization for normal mixture estimation and model-based clustering', *Journal of classification* **24**(2), 155–181.

Gadbury, G. L., Xiang, Q., Yang, L., Barnes, S., Page, G. P. and Allison, D. B. (2008), 'Evaluating statistical methods using plasmode data sets in the age of massive public databases: an illustration using false discovery rates', *PLoS genetics* **4**(6), e1000098.

Gallegos, M. T. and Ritter, G. (2009), 'Trimmed ml estimation of contaminated mixtures', *Sankhyā: The Indian Journal of Statistics, Series A (2008-)* pp. 164–220.

García-Escudero, L. A., Gordaliza, A., Matrán, C. and Mayo-Iscar, A. (2008), 'A general trimming approach to robust cluster analysis'.

García-Escudero, L. A., Gordaliza, A., Matrán, C. and Mayo-Iscar, A. (2010), 'A review of robust clustering methods', *Advances in Data Analysis and Classification* **4**, 89–109.

George, E. (2000), 'The variable selection problem.', *Journal of the American Statistical Association* **95**(452), 1304–1308.

Guyon, I. and Elisseeff, A. (2003), 'An introduction to variable and feature selection', *Journal of machine learning research* **3**(Mar), 1157–1182.

Hastie, T., Tibshirani, R. and Friedman, J. (2017), *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, Springer.

Hennig, C. (2004), 'Breakdown points for maximum likelihood estimators of location–scale mixtures'.

Hennig, C. (2010), 'Methods for merging gaussian mixture components', *Advances in data analysis and classification* **4**, 3–34.

Hinton, G. E. and Salakhutdinov, R. R. (2006), 'Reducing the dimensionality of data with neural networks', *science* **313**(5786), 504–507.

Hosmer, D. (1973), 'A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample', *Biometrics* pp. 761–770.

Hosmer, D. and Dick, N. (1977), 'Information and mixtures of two normal distributions', *Journal of Statistical Computation and Simulation* **6**(2), 137–148.

Huang, J. T. and Hasegawa-Johnson, M. (2009), On semi-supervised learning of gaussian mixture models for phonetic classification, *in* 'Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing', pp. 75–83.

Huberty, C. J. (1975), 'Discriminant analysis', *Review of Educational Research* **45**(4), 543–598.

Hurley, C. B. (2004), 'Clustering visualizations of multidimensional data', *Journal of Computational and Graphical Statistics* **13**, 788–806.

James, G., Witten, D., Hastie, T., Tibshirani, R. et al. (2013), *An introduction to statistical learning*, Vol. 112, Springer.

Japkowicz, N. and Stephen, S. (2002), 'The class imbalance problem: A systematic study', *Intelligent data analysis* **6**(5), 429–449.

Jiang, Z., Zheng, Y., Tan, H., Tang, B. and Zhou, H. (2016), 'Variational deep embedding: An unsupervised and generative approach to clustering', *arXiv preprint arXiv:1611.05148* .

John G, Kohavi R, P. K. (1994), 'Irrelevant features and the subset selection problem', *Machine learning proceedings* pp. 121–129.

Kadhim, R. R. and Kamil, M. Y. (2022), 'Comparison of breast cancer classification models on wisconsin dataset', *Int J Reconfigurable & Embedded Syst ISSN* **2089**(4864), 4864.

Kohavi, R. and John, G. H. (1997), 'Wrappers for feature subset selection', *Artificial intelligence* **97**(1-2), 273–324.

Kohavi, R. et al. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, *in* 'Ijcai', Vol. 14, Montreal, Canada, pp. 1137–1145.

Kostopoulos, G., Karlos, S., Kotsiantis, S. and Ragos, O. (2018), 'Semi-supervised regression: A recent review', *Journal of Intelligent & Fuzzy Systems* **35**(2), 1483–1500.

Kuhn, M. and Johnson, K. (2013), *Applied predictive modeling*, Springer.

Laird, N. (1978), 'Nonparametric maximum likelihood estimation of a mixing distribution', *Journal of the American Statistical Association* **73**(364), 805–811.

Laird, N. M. and Ware, J. H. (1982), 'Random-effects models for longitudinal data', *Biometrics* pp. 963–974.

Lee, D.-H. et al. (2013), Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, *in* 'Workshop on challenges in representation learning, ICML', Vol. 3, Atlanta, p. 896.

Lindsay, B. G. (1995), Mixture models: theory, geometry, and applications, Ims.

Lindstrom, M. J. and Bates, D. M. (1988), 'Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data', *Journal of the American Statistical Association* **83**(404), 1014–1022.

Liu, H. and Setiono, R. (1997), 'Feature selection via discretization', *IEEE Transactions on knowledge and Data Engineering* **9**(4), 642–645.

Maugis, G Celeux, M. M.-M. (2011), 'Variable selection in model-based discriminant analysis', *Journal of multivariate analysis* (102), 1374–1387.

Mazza, A. and Punzo, A. (2020), 'Mixtures of multivariate contaminated normal regression models', *Statistical Papers* **61**(2), 787–822.

McLachlan, G. (1992), *Discriminant analysis and statistical pattern recognition*.

McLachlan, G. and Basford, K. (1988), *Mixture Models: Inference and applications to clustering*.

McLachlan, G. J. (1982), '9 the classification and mixture maximum likelihood approaches to cluster analysis', *Handbook of statistics* **2**, 199–208.

McLachlan, G. J., Bean, R. W. and Jones, L. B.-T. (2007), 'Extension of the mixture of factor analyzers model to incorporate the multivariate t-distribution', *Computational Statistics & Data Analysis* **51**(11), 5327–5338.

McLachlan, G. J. and Ganesalingam, S. (1982), 'Updating a discriminant function on the basis of unclassified data', *Communications in Statistics-Simulation and Computation* **11**(6), 753–767.

McLachlan, G. J., Ng, S.-K. and Bean, R. (2006), 'Robust cluster analysis via mixture models', *Austrian Journal of Statistics* **35**(2&3), 157–174.

McLachlan, G. J. and Peel, D. (2000), *Finite mixture models*, Wiley Series in Probability and Statistics, New York.

McLachlan, G. J., Peel, D. and Bean, R. W. (2003), 'Modelling high-dimensional data by mixtures of factor analyzers', *Computational Statistics & Data Analysis* **41**(3-4), 379–388.

McNicholas (2010), 'Model-based classification using latent gaussian mixture models', *Journal of Statistical Planning and Inference* **140**(5), 1175–1181.

McNicholas, P. D. (2017), *Mixture model-based classification*.

McNicholas, P. D. and Murphy, T. B. (2008), 'Parsimonious gaussian mixture models', *Statistics and Computing* **18**, 285–296.

McNicholas, P. D., Murphy, T. B., McDaid, A. F. and Frost, D. (2010), 'Serial and parallel implementations of model-based clustering via parsimonious gaussian mixture models', *Computational Statistics & Data Analysis* **54**(3), 711–723.

Meek, C. (1997), Graphical Models: Selecting causal and statistical models, PhD thesis, Carnegie Mellon University.

Meng, X.-L. (1994), 'On the rate of convergence of the ecm algorithm', *The Annals of Statistics* pp. 326–339.

Meng, X.-L. and Rubin, D. B. (1993), 'Maximum likelihood estimation via the ecm algorithm: A general framework', *Biometrika* **80**(2), 267–278.

Morris, K., McNicholas, P. D. and Scrucca, L. (2013), 'Dimension reduction for model-based clustering via mixtures of multivariate t-distributions', *Advances in Data Analysis and Classification* **7**(3), 321–338.

Murphy, T. B., Dean, N. and Raftery, A. E. (2010), 'Variable selection and updating in model-based discriminant analysis for high dimensional data with food authenticity applications', *The annals of applied statistics* **4**(1), 396.

Murray, G. and Titterington, D. (1978), 'Estimation problems with data from a mixture', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **27**(3), 325–334.

Naderi, M. N. (2024), 'Clustering asymmetrical data with outliers: Parsimonious mixture of contaminated mean-mixture of normal distributions', *Journal of Computational and Applied Mathematics* **437**.

Neykov, N., Filzmoser, P., Dimova, R. and Neytchev, P. (2007), 'Robust fitting of mixtures using the trimmed likelihood estimator', *Computational Statistics & Data Analysis* **52**(1), 299–308.

Nigam, K., MacCallum, A. and Thrun, S. (1998), *Using EM to classify text from labeled and unlabeled documents*, Carnegie-Mellon University. Department of Computer Science.

Ogrinc, I Košir, J. S.-J. K. (2003), 'The application of nmr and ms methods for detection of adulteration of wine, fruit juices, and olive oil.', *Anal Bioanal Chem* **376**, 424–430.

Peel, D. and McLachlan, G. J. (2000), 'Robust mixture modelling using the t distribution', *Statistics and Computing* .

Punzo, Blostein, M. (2020), 'High-dimensional unsupervised classification via parsimonious contaminated mixtures', *Pattern recognition* **98**.

Punzo, Mazza, M. (2018), 'Contaminatedmixt: an r package for fitting parsimonious mixtures of multivariate contaminated normal distributions', pp. 1–25.

Punzo and McNicholas (2016), 'Parsimonious mixtures of multivariate contaminated normal distributions', *Biometrical Journal* **58**, 1506–1537.

Rachman, S. (2004), 'Fear of contamination', *Behaviour research and therapy* **42**, 1227–1255.

Raftery, A. and Dean, N. (2006), 'Variable selection for model-based clustering', *Journal of the American Statistical Association* pp. 168–178.

Raschka, S. and Mirjalili, V. (2019), *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*, Packt publishing ltd.

Redner, R. A. and Walker, H. F. (1984), 'Mixture densities, maximum likelihood and the em algorithm', *SIAM review* **26**(2), 195–239.

Ripley, B. D. (2007), *Pattern recognition and neural networks*, Cambridge university press.

Ritter, G. (2014), *Robust cluster analysis and variable selection*, CRC Press.

Rizve, M. N., Duarte, K., Rawat, Y. S. and Shah, M. (2021), 'In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning', *arXiv preprint arXiv:2101.06329* .

Russell, S. J., Norvig, P. and Chang, M.-W. (2022), *Artificial intelligence: a modern approach*, fourth / global contributing writers, ming-wei chang [and eight others] edn, Pearson, Harlow, United Kingdom.

Sander, J., Ester, M., Kriegel, H.-P. and Xu, X. (1998), 'Density-based clustering in spatial databases: The algorithm gdbscan and its applications', *Data mining and knowledge discovery* **2**, 169–194.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The annals of statistics* pp. 461–464.

Scrucca, L. (2010), 'Dimension reduction for model-based clustering', *Statistics and Computing* **20**, 471–484.

Seok, K. (2014), 'Semi-supervised regression based on support vector machine', *Journal of the Korean Data and Information Science Society* **25**(2), 447–454.

Sexton, J. and Swensen, A. R. (2000), 'Ecm algorithms that converge at the rate of em', *Biometrika* **87**(3), 651–662.

Singh, A., Nowak, R. and Zhu, J. (2008), 'Unlabeled data: Now it helps, now it doesn't', *Advances in neural information processing systems* **21**.

Smith, A. and Makov, U. (1978), 'A quasi-bayes sequential procedure for mixtures', *Journal of the Royal Statistical Society: Series B (Methodological)* **40**(1), 106–112.

Sokolova, M. and Lapalme, G. (2007), Performance measures in classification of human communications, *in* 'Advances in Artificial Intelligence: 20th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2007, Montreal, Canada, May 28-30, 2007. Proceedings 20', Springer, pp. 159–170.

Sokolova, M. and Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Information processing & management* **45**(4), 427–437.

Spadaro, D., Lorè, A., Garibaldi, A. and Gullino, M. L. (2010), 'Occurrence of ochratoxin a before bottling in doc and docg wines produced in piedmont (northern italy)', *Food Control* **21**(9), 1294–1297.

Steinley, D. (2003), 'Local optima in k-means clustering: what you don't know may hurt you.', *Psychological methods* **8**(3), 294.

Steinley, D. and Brusco, M. J. (2008), 'Selection of variables in cluster analysis: An empirical comparison of eight procedures', *Psychometrika* **73**, 125–144.

Tibshirani, R., Walther, G. and Hastie, T. (2001), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2), 411–423.

Titterington, D. (1976), 'Updating a diagnostic system using unconfirmed cases', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **25**(3), 238–247.

Titterington, D. M. (1990), 'Some recent research in the analysis of mixture distributions', *Statistics* **21**(4), 619–641.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985), *Statistical analysis of finite mixture distributions*, Wiley, Chichester.

Tukey, J. W. (1960), 'A survey of sampling from contaminated distributions', *Contributions to probability and statistics* pp. 448–485.

Von Weinen, M. D. z. S. (1986), 'Multivariate data analysis as a discriminating method of the origin of wines', *Vitis* **25**, 189–201.

Wehrens, R. (2011), *Chemometrics with R*, Vol. 3, Springer.

Weiss, G. M. and Provost, F. (2001), The effect of class distribution on classifier learning: an empirical study, Technical report, Rutgers University.

Wolfie, J. (1970), 'Pattern clustering by multivariate mixture analysis', *Multivariate behavioral research* **5**(3), 329–350.

Wood, S. N. (2017), *Generalized additive models: an introduction with R*, chapman and hall/CRC.

Xie, J., Girshick, R. and Farhadi, A. (2016), Unsupervised deep embedding for clustering analysis, *in* 'International conference on machine learning', PMLR, pp. 478–487.

Yan, Zhou, P. (2017), 'Gaussian mixture model using semisupervised learning for probabilistic: fault diagnosis under new data categories', *IEEE Transactions on Instrumentation and Measurement* **66**(4), 723–733.

Yang, T. and Priebe, C. E. (2011), 'The effect of model misspecification on semi-supervised classification', *IEEE transactions on pattern analysis and machine intelligence* **33**(10), 2093–2103.

Zhu, L., Gamez, G., Chen, H., Chingin, K. and Zenobi, R. (2009), 'Rapid detection of melamine in untreated milk and wheat gluten by ultrasound-assisted extractive electroscopy ionization mass spectrometry (eesi-ms)', *Chemical communications* (5), 559–561.

Zhu, X. and Goldberg, A. B. (2022), *Introduction to semi-supervised learning*, Springer Nature.

Śmieja, Wołczyk, T. G. (2020), 'Semi-supervised gaussian mixture autoencoder', *IEEE transactions on neural networks and learning systems* **32**(9), 3930–3941.