



Faculty of Sciences
Department of Computer Science & Information Systems

Outlier Management Based On Support Vector Machine

by

Kelebogile Thutlwe

Reg. No: 18100087

BSc (Computing with Finance), University of Botswana, Botswana

E-mail: tk18100087@studentmail.biust.ac.bw

A Thesis Submitted to the Faculty of Sciences in Partial Fulfilment of the Requirements
for the Award of the Degree of Master of Science in Computer Science of BIUST

Supervisors:

Dr. Thabiso Maupong

Department of Computer Science & Information Systems,

Faculty of Sciences, BIUST

E-mail: maupongt@biust.ac.bw

Dr. Nkumbuludzi Ndwapi

Department of Mathematics & Statistical Sciences,

Faculty of Sciences, BIUST

E-mail: ndwapin@biust.ac.bw

May 30, 2023

Declaration



POSTGRADUATE SCHOOL

Declaration Regarding the Work and Copyright

DECLARATION REGARDING THE WORK AND COPYRIGHT

Candidate: _____ KELEBOGILE THUTLWE _____

Student ID: _____ 18100087 _____

Thesis Titled: _____ OUTLIER MANAGEMENT USING SUPPORT VECTOR MACHINES _____

I, KELEBOGILE THUTLWE, certify that the Thesis is all my own original work and that I have not obtained a degree in this University or elsewhere on the basis of any of this work.

(If the thesis is based on a group project, then the student must indicate the extent of her / his contribution, with reference to any other theses submitted or published by each collaborator in the project, and a declaration to this effect must be included in the thesis)

This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright and Neighbouring Rights Act, Act. No. 8 of 2000 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Postgraduate School, on behalf of both the author and the BIUST.

Signed: _____  _____ Date: _____ 24/01/2023 _____

Primary Supervisor : _____ DR. THABISO MAUPONG _____

I, the Candidate's **Primary Supervisor**, hereby confirm that I have inspected the above titled thesis and, to the best of my knowledge, it is based on the original work of the candidate.

Signed: _____  _____ Date: _____ 24 Jan 2023 _____

Abstract

Outliers describe data points that deviate from the rest of a dataset. These deviations may be caused by the following but not limited to; errors during data collection, instrumentation errors and faults in data entry. These faults and errors make it difficult to capture and accurately model the patterns that are inherent in the data. On the other hand, outliers may be an indication of new discoveries in a field, calling for more research on why or how these outlying data points came about. Considering the potential benefits of outliers as well as the problems they may pose, this work focuses on the management of outliers in multivariate datasets with the primary focus in classification problems. In this work, an algorithm to relabel outliers is developed. Support Vector Machine (SVM) is used to detect outliers given its effectiveness in dealing with high dimensional data. After detection, the outliers are relabelled using the developed technique. This technique, measures how far off the outliers are in relation to respective class centroids. Then, based on the comparison between the measured distances, the outliers are relabelled. To evaluate the effectiveness of the developed technique, we use discrimination techniques such as Fisher's Linear Discriminant Analysis (FLDA), Gaussian Mixture Model (GMM) and the unsupervised clustering technique to highlight the extent of discrimination. The results show a significant improvement and in some cases obtain a 100% classification on evaluation using the Adjusted Rand Index (ARI) metric.



Acknowledgements

I would like to thank my supervisors Dr. Maupong and Dr. Ndwapi for their guidance and unwavering support during my study period and the opportunities they exposed me to. I would also like to thank my family and friends for encouraging me to work harder and to not give up when I hit stumbling blocks along the way. Last but not least I would like to thank God for giving me strength to carry on and see this work through to completion.



Contents

Abstract	ii
1 Introduction	2
1.1 Outliers	2
1.2 Classification	3
1.3 Outlier Management	4
1.4 Problem Statement	6
1.5 Contributions of this research work	7
1.6 Thesis Outline	7
2 Background	9
2.1 Support Vector Machine	9
2.1.1 Separating Hyperplane	9
2.1.2 Maximal Margin Classifier	9
2.1.3 Support Vector Classifier	10
2.1.4 Support Vector Machine	11
2.1.5 Latest Work on Support Vector Machines and Outliers	12
2.2 Clustering	14
2.2.1 KMeans Clustering	14
2.2.2 Hierarchical Clustering	14
2.3 Fisher's Linear Discriminant Analysis	15
2.3.1 Rand Index (RI)	16
2.3.2 Adjusted Rand Index (ARI)	16
2.4 Mixture Models	17
2.4.1 Parameter Estimation	18
2.4.2 Maximum Likelihood Estimation	18
2.4.3 Maximum Likelihood Estimation for Gaussian Distribution	18
2.4.4 Log-Likelihood Function of the Gaussian distribution	19
2.4.5 Maximum Likelihood Estimation for the Gaussian Mixture Model	19
2.4.6 Expectation Maximization Algorithm	20
2.5 Relabelling Techniques	21
2.6 Techniques used to measure distance of support vectors from the separating hyperplane	22
2.6.1 Mahalanobis distance	23
2.6.2 Summary	24
3 Proposed Relabelling Technique	25
3.1 Constructing the SVM	25
3.2 Training the SVM model	26
3.3 Relabelling algorithm	28
3.4 Step 1: LDA on original datasets	30
3.5 Step 2: LDA modified by the support vectors	30
3.6 Gaussian Mixture Model for the dataset scenarios	31
3.7 Summary	33
4 Results and Discussions	34

4.1	Datasets	34
4.1.1	Iris	34
4.1.2	Breast Cancer	36
4.1.3	Wine	37
4.2	Discussion	38
4.2.1	Summary of the performance of the SVM on various datasets	40
4.2.2	Performance scores of the LDA on different datasets	40
4.2.3	Performance scores of the LDA on the GMM of the different datasets	41
4.3	Linear Discriminant Analysis on a Gaussian Mixture model on the datasets	41
5	Conclusions and Future work	43
5.1	Future work	43



List of Figures

2.1	Obtained from James et al. [2013]-Observations 11 and 12 on the second grid are on the wrong side of the hyperplane as well as the margin and have an epsilon $\epsilon_1 > 1$ as per the optimization problem described in the equation above.	10
2.2	Obtained from Yang et al. [2007] - Different decision boundaries in the presence of outliers as opposed to when they are absent.	12
3.1	Distance of the support vectors from the centroid and centroid from the separating hyperplane on a hard margin SVM	28
3.2	Distance of the support vectors from the centroid and centroid from the separating hyperplane on a soft margin SVM	28
3.3	Positions of the support vectors from the separating hyperplane and as shown by Mahalanobis distance	30
3.4	Process flowchart	32
4.1	The scatter plot for the classes setosa, versicolor and virginica	35
4.2	Hyperplane separating between setosa and virginica	35
4.3	The scatter plot for the classes setosa, versicolor and virginica on the attributes, sepal length and sepal width	35
4.4	The scatter plot for the first three attributes mean radius, mean texture and mean perimeter	36
4.5	The scatter plot for the first two attributes worst area and mean area	37
4.6	The 3D scatter plot for the first three attributes alcohol, malic acid and ash	37
4.7	The 2D scatter plot for the first two attributes alcohol, malic acid and ash	37
4.8	The scatter plot for the first two attributes length, left and right edges	38
4.9	The scatter plot for the first two attributes length of the bill and left edge	38
4.10	Elbow method to determine the number of clusters for a KMeans clustering on the Iris dataset	41

List of Tables

4.1	Table showing the performance of the SVM on different kernels at different training to test set ratios	39
4.2	Table showing the mean ARI scores performed on different datasets at different training to test set ratios	39
4.3	Table showing LDA results for the GMM performed on datasets	40
4.4	Table showing the silhouette scores of the performance of KMeans clustering on GMM's	42



1. Introduction

Data processing and analysis play a pivotal role in various disciplines and or applications including but not limited to; data mining, data science, data driven control, pattern recognition, fraud detection and intrusion detection. Data is used to infer knowledge in the form of data models otherwise termed as generalizations, in order to be able to predict or determine what will happen to new data points in comparison to those used to generalize or come to the reached conclusions. It is essential then, that the data that is used to generalize is reliable. Therefore, it is crucial that those data points that don't belong or may seem to not belong to any form of discrimination be it classification or clustering are handled with care. These data points are termed as *outliers*.

There are different types of outliers;

- Univariate outliers; outliers appearing in a single feature space, for example, when capturing the ages of a population, those who have lived longest will appear as outliers or when capturing the heights of men, the tallest and shortest men in the distribution will stand out on the curve or the graph used to visualize the patterns in the dataset.
- Bivariate outliers; outliers present in a two dimensional space where there is some relationship between the features for example age and height of students. These outliers are as well easy to point out especially with the use of data visualization tools and methods such as bar charts and scatter-plots.
- Multivariate outliers; Outliers present when datasets have over two features. It becomes more difficult here to point out these outlying points because of the complexity of high dimensional spaces, making it important to develop models that are better able to capture these outliers.

While various disciplines use different means to detect and treat outliers, it is a common practice to remove outlying data points. In most cases, the mechanism used to remove outliers is more in relation to the dataset being dealt with, for instance, an English physicist Rayleigh is said to have discovered a new gas then, by having observed the differences in the measurements of the gas densities. He is said to have discovered that atmospheric nitrogen gas always recorded a relatively greater density than nitrogen obtained from chemical compounds. This observation implied that there was a denser gas present, and that was how the gas argon, was discovered. A discovery that led to the Nobel prize award for Physics in 1904, see [Cheng \[2002\]](#). This factor, implies that outliers should not just be easily dealt away with unless there has been more attempt to investigate the origin and reason for encountering outliers before doing away with them.

1.1 Outliers

This section gives an insight on what constitutes an outlier and the various methods for detection and management techniques. [Hawkins \[2013\]](#) defines outliers as observations which deviate from the other observations raising doubt that they have been produced by a different approach.

[Cheng \[2002\]](#), [Osborne and Overbay \[2004\]](#) describe outliers as abnormal points in data. These authors further point out that depending on the data, outliers may give analysts false information or conversely, useful information that could lead to new discoveries. Data riddled with outliers comes with the challenge of determining how much of the outliers to reject. Furthermore, this

data point rejection should be done carefully to avoid loss of useful information as this may compromise analysis of the data in the end.

It is possible that errors in data collection and entry are not the cause of outliers. In this scenario, it cannot be conclusive that an outlier is a bad point. Outliers could be indicative of novelty in the field, requiring further research into the new discovery. In the literature it has been shown that other than novelty, outliers otherwise play an important role in various applications such as network intrusion, fault detection systems, credit card fraud, weather prediction, medical diagnosis, to name a few, see [Divya and Babu \[2016\]](#), [Cheng \[2002\]](#).

Statistically, outliers depend on an underlying distribution. According to the authors [Last and Kandel \[2001\]](#) outliers are values that fall far from the centre of the distribution on either direction. The authors further describe outliers as not only pertaining to distance but also to frequency of occurrence. Therefore, this suggests that an outlier is a data point which will be encountered in the dataset less frequently in comparison to other data points. This definition shows that the inconsistencies in the data cannot be straight away considered good or bad unless in extreme cases where the data being handled is not expected to yield such recorded points. While in some cases it may be clear that there may have been errors during data collection or entry, sometimes there is need to investigate why or what could be the reason for encountering an outlier point.

In classification problems, outliers are considered as test instances that have not been classified into either one of the learnt classes. This means, data points that do not fall in either one of the categories that the classifier has learned during training may be considered outliers, see [Upadhyaya and Singh \[2012\]](#). This is the definition adopted in this work. Its application is later seen by use of support vectors in a Support Vector Machine.

Given the various definitions of outliers as may be applied in various disciplines and having identified a more suitable definition for this work, it is therefore more important to understand what classification is as a Machine Learning technique that will be used in this work. The next section defines classification and the problems that surround it.

1.2 Classification

Data is dealt with in various ways and to derive different patterns some quantitative, for example, as in regression problems. However, at times interest is not in deriving or inspecting the quantitative aspect of data but in its *qualitative* responses. That is to say, data is labelled or categorized into groups otherwise termed as classes. In this regard, *classification* is then defined as a supervised learning technique according to [James et al. \[2013\]](#) and [Kotsiantis et al. \[2007\]](#), where categorical responses are predicted upon learning some patterns by training a model from the features of a given dataset. Predicting a category or a class is categorizing an observation as belonging to one of the learnt classes or categories.

[Michie et al. \[1994\]](#) also shares the same sentiments as [James et al. \[2013\]](#) and [Kotsiantis et al. \[2007\]](#) that a classification method uses a set of data to determine patterns that are known prior. [Michie et al. \[1994\]](#) otherwise terms classification as a pattern recognition or discrimination so as to distinguish it from unsupervised learning where the classes are not known prior. It is also important to know the approaches to classification being supervised and unsupervised by which more attention is paid to the former in this thesis.

It is easier to classify data points when resultant data models have been trained on balanced data, that is, where the classes have equal representation (equal number of observations per class), but in application data is often unbalanced. This is one of the limitations with classification. The authors [Ali et al. \[2015\]](#), [Barandela et al. \[2003\]](#), [Mollineda et al. \[2007\]](#), [Kotsiantis et al. \[2006a\]](#)

all agree that most classification problems suffer from a problem of imbalance of classes where one class may have less representation in the dataset. Ali et al. [2015] explains that small sample size in training sets also contributes to lower classification accuracy and that this problem also contributes to the problem of class overlapping where the degree of separability of the classes is not clear.

Kotsiantis et al. [2006a], Kotsiantis et al. [2006b] and Ali et al. [2015] all agree that sampling whether under-sampling or oversampling may come in as a data level approach for handling the problem of imbalance of classes. According to Kotsiantis et al. [2006b], random sampling, where part of the dataset is selected randomly or stratified sampling, where observations from a minority class are selected with greater frequency in order to balance the distribution may be used. Stratified sampling according to Sharma [2017] applies when the class values are not distributed evenly in the training sets. Later on, some of these techniques used to deal with the class imbalance problem are assimilated because most of the datasets are not balanced as will be noticed in the latter chapters of this thesis.

The discussions outlined above give more insight on what an outlier is as may be encountered in different fields of study. In light of what outliers are and with more emphasis on outliers in classification, it is therefore important to investigate how conclusions on what constitutes an outlier and what does not can be arrived at. *Outlier detection* in this light, is the next item of discussion before getting to the management of outliers towards better parameter estimates and effective data reflective modelling.

1.3 Outlier Management

In this section, the different techniques and approaches for managing outliers from different domains and applications are explored based on existing literature. According to Xu et al. [2018], outlier detection is used in areas such as clustering, pattern classification and decision making. While it may be easier to identify outliers in low dimensional spaces, for high dimensional data, distance is said to lessen significantly making it difficult to detect outliers using the distance based techniques. Xu et al. [2018] discussed the following techniques that are aimed at dealing with high dimensional data; Neighbour ranking based methods which use the rank of neighbours as a proximity measure as opposed to using distance. Subspace learning based methods which detect outliers using only appropriate attributes because not all attributes contain significant information. Ensemble learning based methods, since it is difficult to use only one technique to detect outliers given the complexity of data in real life, ensembles are necessary here to further detect outliers in multiple subspaces.

On the detection of outliers, Acuña and Rodriguez [2004] vary the detection according to the subspaces that the outliers are found in. Acuña and Rodriguez [2004] explains univariate outliers, which are usually depicted in a one dimensional space. These outliers are detected using exploratory data analysis tools such as box plots which are dependant on the distribution. The box-plot is able to detect mild and extreme outliers by the interquartile ranges the outliers fall in.

Acuña and Rodriguez [2004] describe multivariate outliers to be detected in various ways including; statistical based methods, detection by clustering, distance based and density based methods. A combination of these techniques such as distance based outlier detection, detection by cluster analysis were used to demonstrate their influence to parametric estimation.

According to Aggarwal [2016], outlier detection algorithms such as the score and binary labels can be used to measure the outlierness (mild or extreme outliers) of data points. Scores measure the outlierness of a point and binary labels determine whether a data point is an outright outlier

or not. [Aggarwal \[2016\]](#) describes different methods of outlier detection models including, probabilistic and statistical models, linear models, proximity based, high dimensional, supervised, categorical, text and mixed attribute data, time series and multidimensional streaming, discrete sequences, spatial data, graphs and networks models. However, given the many domains outlined in the book [Aggarwal \[2016\]](#) interest is in supervised outlier detection methods with more emphasis on the case of classification.

The authors [Mandhare and Idade \[2017\]](#) and [Kamble and Doke \[2017\]](#) categorize outlier detection techniques as described below;

- *Cluster based*: Combines data objects according to the information received from the data which consists of the object relationship and grouping similar objects. Example models that use this technique include; K-Means, Expectation Maximization(EM) and other techniques.
- *Distance based*: The parameters and distance between the data points are given in order to detect outliers. The distance between the data points (usually the Euclidean distance) is calculated and points furthest from others are deemed as outliers and those much closer as normal data points.
- *Density based*: Each data point is referred to as a local outlier factor (LOF) where LOF is the ratio between the local density of an object and the average of its k nearest neighbors. An object's local outlier score is checked and those having the highest LOF are considered as outliers and those with low LOF as normal data.
- *Statistical based*: Mean values are calculated and the data point assessed against these mean values as in the classic case of the Gaussian distribution.
- *Deviation based*: A data point's properties are assessed for conformity to the rest.

The author [Escalante \[2005\]](#) describes outlier detection in statistical learning, to use properties of data distribution. The properties include probabilistic and Bayesian methods to model inconsistencies in data distributions. [Escalante \[2005\]](#) further states that the limitation with these detection schemes is that they tend to support univariate data and multivariate samples with only a few dimensions. Processing time becomes a limitation as distance based techniques and functions showing dissimilarity in the data attributes require parameters that are difficult to set due to high dimensional spaces. According to [Escalante \[2005\]](#), at times ensembles (combinations of classifiers) are successfully used in classification problems to identify mislabelled points.

Outliers may be analysed according to which field of study the data is being dealt with, see [Cheng \[2002\]](#). This is due to the fact that outliers in various domains may be caused by different factors. For instance, an outlier in banking information may imply that there was theft in an Automatic Teller Machine (ATM) where a large unusual ATM cash withdrawal were discovered.

[Last and Kandel \[2001\]](#) suggest careful examination of scatter plots to detect outliers. They also mentioned manual detection of which is said to be bound to the limitation of subjectivity and poor scalability. They suggest that a quantitative approach consisting of boxplots that use the median of the data to be used as a possible solution.

In order to improve classification accuracy, [Smith and Martinez \[2011\]](#) suggest that outliers or misclassified points should be removed entirely from the training set. They use a method called Preprocessing Instances that Should be Misclassified (PRISM) to identify and remove instances that should otherwise be misclassified and compare it against a few existing outlier detection and noise reduction methods. In relation to outliers, [Smith and Martinez \[2011\]](#) observe that there is no universal agreement on what constitutes an outlier and that there is no clear distinction between outliers and noise from exceptions. Therefore, their argument alludes to the fact that

since there is no conformity on what constitutes an outlier, important data points may well be lost during the data cleaning process when removing the outliers from data.

Hodge and Austin [2004] discuss outlier detection approaches that are suited for unsupervised learning where there is no prior knowledge of the dataset. The data is processed as a static distribution which singles out the most remote points as outliers. They further described another approach to detect outliers in supervised classification that requires prior knowledge of the data labels to infer knowledge on which is normal and which is abnormal. Furthermore, they described a technique cognate to the semi supervised detection which usually models novelty by modelling what is normal and in few instances, abnormal cases.

Ndwapi [2018], suggest creating a robust discriminant analysis model that uses the SVM and the Mixture Discriminant Analysis (MDA). The proposed model makes use of the support vectors which are detected when training the SVM. The misclassified data points (support vectors) are assigned to the classes that they find themselves in hence improving parameter estimation. The authors suggest that using Support Vector Machines to detect outliers will improve parameter estimates in Discriminant Analysis and hence improving the classification of unlabelled observations. Mostly, this work aims to achieve a similar outcome assuming that the underlying distribution of the classes are not different.

Even though most authors support the removal of outliers as a data preprocessing technique, the limitation associated with this approach is the loss of data. Data in various areas is considered to be quite valuable. Analysts in these various areas should strive to not lose data points. This is because the data points lost may contain information that is crucial to the successful modelling of the data. This may lead to data models performing adversely due to loss of information.

An alternative way of managing outliers in classification is proposed in this work. In this proposed method, Support Vector Machine (SVM) which solely depends on the data points deemed as outliers (support vectors) to construct the classifier is used. The SVM is used to detect outliers when training the model. Once detected, the outliers/support vectors are relabelled relative to their distances away from the separation hyperplanes. By having relabelled the outliers, a new working dataset is extracted. With this new dataset, the labels are removed from the dataset and advertently, the dataset trained using LDA. It is expected that the performance scores measured using the ARI metric show that relabelling, in comparison to removing outliers yields as much good results. This is evidence enough that instead of immediately removing data points deemed as outliers(bad outliers), finding a way to lessen their impact on parametric estimation in support to losing as minimal data points as possible is an equally befitting approach.

SVMs have been identified as a rigorous technique for outlier detection given their sensitivity to outliers. As seen in Thongkam et al. [2008], the SVM was used to identify misclassified instances and those outliers were removed from the samples. SVMs have also been used in wireless sensor networks to detect outliers as evident in Mohamed and Kavitha [2011]. The SVM algorithm has also been used in computer security for anomaly detection as may be seen in Hu et al. [2003] to detect outliers. While SVMs have been widely used to detect and remove outliers as an impact mitigation measure, SVMs have not been used to relabel outliers as done in this work, neither has a combination of two outlier detection techniques, SVM and Mahalanobis distance been used to treat outliers in the manner used in this research. This constitutes a unique contribution of this work in this field of research.

1.4 Problem Statement

The discussions on the treatment of outliers in classification problems show that the general or common phenomenon towards handling outliers is removing them, a phenomenon that threatens

the quality of modelling because data is being lost. The aim is to develop data reflective models that generalise well on new data and so removing some points loses the information that is in these data points which may well help categorise new points. Based on this aspect, this research work investigates the impact of relabelling outliers, on modelling compared to the common approach of removing outliers. It is in this light that in the next section, the aim and contributions of this research work are discussed.

1.5 Contributions of this research work

1. The primary aim of this research was to investigate and compare the impact on modelling, of relabelling outliers as opposed to removing outliers particularly in classification problems.

To achieve this primary objective, a breakdown of the research into objectives. The objectives are as follows;

- i Develop a Support Vector Machine to detect outliers in classification problems with the help of an effective margin optimization technique which uses Lagrange multipliers.
- ii Develop an algorithm to measure the distances of the support vectors from the margins using vector norms, Euclidean distance and Mahalanobis distance.
- iii Provide a relabelling criteria to decide which support vectors to relabel based on their distance away from the margins. The algorithm is also used to remove some of the support vectors in order to compare performance when removing outliers as opposed to when relabelling outliers.
- iv Evaluate how well a classification algorithm performed using more discrimination techniques including clustering methods and Fisher's Linear Discriminant Analysis algorithm on the dataset scenarios as well as on the mixture models of these scenarios where a scenario constitutes; an original dataset that has not been treated for outliers by relabelling or removing, a dataset for which the outliers had been relabelled and a dataset for which the outliers had been removed.

1.6 Thesis Outline

The thesis outline is as follows;

- Chapter 2, looks into the tools used in the thesis. The Chapter gives details of what constitutes a support vector and how they can further be associated with outliers. A review of Fisher's Discriminant Analysis with a focus on Linear Discriminant Analysis is presented. A brief description of Clustering and its performance measures, the Rand Index and The Adjusted Rand Index is included in the Chapter. The Chapter also has parameter estimation of Gaussian Mixture Models and the contribution of the EM algorithm in deriving the estimates of the likelihood function when there are latent variables.
- Chapter 3, gives an overview of process flow of work towards attaining the set objectives. It gives a discussion of how a dataset was derived where outliers are relabelled as well as a dataset where outliers are removed after training the SVM model. A discussion of the algorithm used to determine which support vectors to relabel is also given.
- Chapter 4, is a discussion of the results of the study, firstly datasets that were used are described. A discussion of the problems that were encountered in carrying out these experiments and how they were addressed of course having explored the possibilities that could have been adopted. A discussion on the work that has already been done in summary, the

measurement criteria that are currently used especially the distance based techniques in classification problems with inferences from the literature.

- Chapter 5 addresses the conclusions on what the study has achieved as well as room for improvement in the methodology adopted given more time paving way for future work on the research.



2. Background

In this Chapter, the tools and concepts that are used throughout the thesis are discussed. The SVM as a supervised classification technique, clustering as an unsupervised learning technique, Linear Discriminant Analysis (LDA) to maximize variability between classes, Adjusted Rand Index (ARI) as an evaluation metric for LDA, existing relabelling mechanisms and distance measures used to find how far off the support vectors are from the separating hyperplanes. An insight on distance measures in multivariate data as well as relabelling techniques for misclassified points is given.

2.1 Support Vector Machine

A *Support Vector Machine* (SVM) is a supervised learning classification technique (classifier) conventionally defined by a separating hyperplane. In other words, given labelled training data a case of supervised learning, the algorithm outputs an optimal hyperplane which categorizes new examples [James et al. \[2013\]](#), [Gunn et al. \[1998\]](#).

2.1.1 Separating Hyperplane

The separating hyperplane is defined as follows.

Let X be an $n \times p$ data matrix where n is the number of observations and p is the number of entries under observation $x_1, \dots, x_n$ and that these are binary observations such that $y_1, y_2, \dots, y_n \in \{-1, 1\}$ where 1 represents one class and -1 represents another class.

Let $\beta_0, \beta_1, \dots, \beta_p \in \mathbb{R}$, then for $y_i = 1$,

$$(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) > 0 \quad (2.1)$$

and for $y_i = -1$,

$$(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) < 0. \quad (2.2)$$

Now to generalise Eq. (2.2) for a case where one has p variables and $y_i \in \{-1, 1\}$, then ;

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) > 0 \quad (2.3)$$

for $i = 1, 2, \dots, n$

2.1.2 Maximal Margin Classifier

It is not always the case that there is one separating hyperplane. Therefore, one that best separates the data such that the data points are equi-distant from the hyperplane is identified. Such a separating hyperplane is called the *maximal margin classifier* which finds the hyperplane that gives the largest minimum distance (M) to the training examples asserting confidence that observations are correctly classified.

Let $\beta_0, \beta_1, \dots, \beta_p \in \mathbb{R}$, then to find the maximal margin classifier, one solves the following.

$$\begin{aligned}
& \text{maximize } M_{\beta_0, \beta_1, \dots, \beta_p} \\
& \text{subject to } \sum_{j=1}^p \beta_j = 1 \\
& y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \dots, n.
\end{aligned} \tag{2.4}$$

2.1.3 Support Vector Classifier

There are instances when observations are not separable by a hyperplane. In such cases some observations are allowed to be incorrectly classified as a trade off for better robustness of the classifier to observations and to classify most observations properly. That is to say there will not be any best optimal hyperplane if the aim is to correctly classify the instances and so, some extremely off data points are allowed to be in the wrong class they find themselves in. To do that, introduce what is called *slack variables* $(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$ to Eq. (2.4). These slack variables allow violations to the margins, that is they allow observations to be misclassified;

$$\text{maximize } M \text{ for } \beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n \tag{2.5}$$

under the constraints

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{2.6}$$

$$\epsilon_i \geq 0 \text{ and } \sum_{i=1}^n \epsilon_i \leq C \tag{2.7}$$

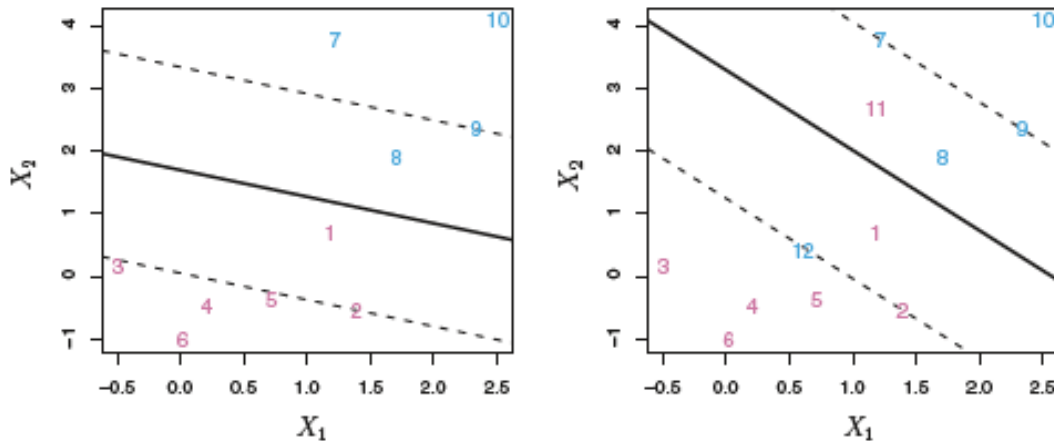


Figure 2.1: Obtained from [James et al. \[2013\]](#)-Observations 11 and 12 on the second grid are on the wrong side of the hyperplane as well as the margin and have an epsilon $\epsilon_i > 1$ as per the optimization problem described in the equation above.

Observations that lie on the margins or the hyperplane or on the wrong sides of the optimal hyperplane are the support vectors as shown in Figure 2.1. These support vectors influence the separating hyperplane and a shift in one of support vectors may greatly affect the SVM classifier resulting in over fitting of the training data. Therefore, a support vector classifier or a soft margin classifier makes it worthwhile to misclassify a few training observations in order to do a better job in classifying the remaining observations.

The aim is to maximize the margin M given $\beta_0, \beta_1, \dots, \beta_p$ and $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ where the summation of the β 's squared should be equal 1 and the regression estimators should be as large as possible. The $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ should be greater than zero but should be well within the budget tuning parameter C which is the maximum allowable number of violations to the margins.

In cases where the data is not separable instead of identifying the largest possible margin such that all the data points are not only on the correct side of the hyperplane but the margin as well, it allows some data points to be on the incorrect side of the margin or even the margin giving us a support vector classifier. The margin is soft because it can be violated by some of the training observations. What is done is introducing the epsilons otherwise known as the slack variables which allow violations to the margin and the tuning parameter C which is the maximum number of violations to the hyperplane that may be allowable chosen by cross validation.

However, there may be instances where the data being dealt with is not separable by a linear hyperplane. This scenario may be tackled by enlarging the feature space which in higher dimension proves to be complex and computationally expensive. The Support Vector Machine in this regard extends on to the support vector classifier to enlarge the feature space through the use of kernels.

2.1.4 Support Vector Machine

In trying to mitigate the complexity problem that comes with enlarging the feature space, the Support Vector Machine is constructed as a solution to this problem. An SVM aims to create decision boundaries that are not linear to solve cases of data that is inseparable by linear boundaries and this is done by the use of *kernels*.

Kernels are a computationally efficient approach to enlarge the feature space by way of using inner products of the data points as opposed to the data points themselves. The inner product of two distinctive observations x_i, x'_i is given by:

$$\langle x_i, x'_i \rangle = \sum_{j=1}^n x_{ij} x'_{ij} \quad (2.8)$$

where n is the number of observations.

Based on the description of inner products, a kernel function K depicting the similarity between two observations is defined as;

$$K(x_i, x'_i) = \sum_{j=1}^p x_{ij} x'_{ij} \quad (2.9)$$

The linear support vector classifier is presented as;

$$f(x) = \beta_0 + \sum_{i=1}^n \alpha_i \langle x_i, x'_i \rangle \quad (2.10)$$

where β_0 is a constant and the number of α_i parameters is n .

Some of the commonly used kernels include the linear kernel, polynomial kernel, and the radial kernel. The linear kernel is the standard SVM which is equivalent to the support vector classifier shown in Eq. 2.10.

Other kernels include the polynomial and the radial which best works in quite high dimensional spaces where computations are complex. The following functions $f(x)$ in Eq. (2.11) and K_r in Eq. (2.12) show the polynomial and radial kernel respectively, where;

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K_p(x, x_i) \quad (2.11)$$

where S is the support vector indices and K_p is a kernel function.

$$K_r(x_i, x'_i) = \exp \left\{ -\gamma \sum_{j=1}^n (x_{ij} - x'_{ij})^2 \right\} \quad (2.12)$$

where x is a real vector and γ is a positive constant, where terms x_{ij} and x'_{ij} represents two distinct observations.

2.1.5 Latest Work on Support Vector Machines and Outliers

According to the authors of Yang et al. [2007], when dealing with SVM outliers are identified as the support vectors which make up the SVM. It is due to this instance that minor shifts in the support vectors cause shifts in the optimal hyperplane showing the sensitivity of the SVM to the outliers. Figure 2.2 illustrates how shifts in support vectors may greatly affect the separating hyperplanes and subsequently classification of data points.

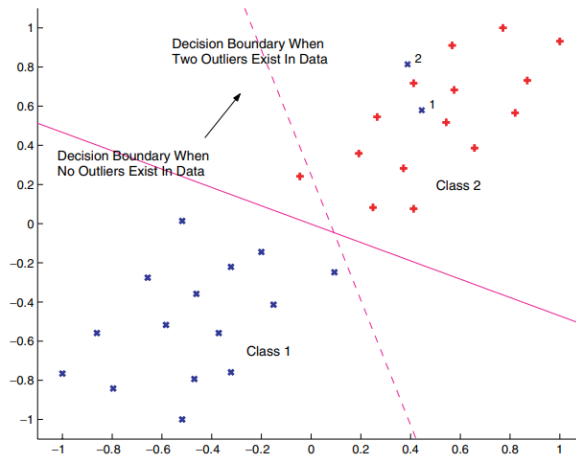


Figure 2.2: Obtained from Yang et al. [2007] - Different decision boundaries in the presence of outliers as opposed to when they are absent.

In Figure 2.2, the solid line shows the optimal hyperplane when there are no outliers in the dataset, therefore perfectly classifying the classes and the dotted line represents the optimal hyperplane in presence of outliers which are the blue points 1 and 2 that find themselves amidst the red points (Class 2) and the red point that finds itself on the other side of the optimal hyperplane among the blue points (Class 1). This scenario indicates that indeed the optimal hyperplane is sensitive to outliers as the presence of outliers coerces the hyperplane to shift so that there is always maximal distance from the data points.

Literature on data classification suggests some of the following techniques to deal with outliers especially when dealing with SVMs;

The authors Mohammadi and Sarmad [2019] state that outlier detection is a pre-training task that in the end a researcher may opt to remove the identified outliers or use a robust SVM

to intercept reducing the accuracy of classification. The authors propose using a Minimum Covariance Determinant Estimator (MCD) to detect outliers for SMVs. In this technique, they suggest that three things have to be present in order to construct the MCD classifier based on the author [Debruyne et al. \[2009\]](#)'s outlier map. They outline; classifying function, the cut-off point and the measure of outlierness that is how far off a point deviates from what is expected-in this case, the cut-off point. In their proposed model, they suggest using a an MCD- based robust Mahalanobis distance (RMD)- which is the difference between two points and for those with a higher RMD than the cut-off will be deleted and hence the effect of outlying points on SVM is reduced.

The authors of [Kaiser \[2013\]](#) discuss a class of robust SVM with a reduced sensitivity to outliers by using an implicit outlier filtering method, and in this technique, they also use the SVM to identify and remove outliers relative to their positions from the separating hyperplane. The authors say;

“We are especially interested if the additional parameter which controls the removal of outliers can be estimated from a training set which is contaminated by outliers.”

The preceding discussions show the persistent pattern of removing outliers as a data preprocessing activity in addition to some, discussed in Chapter 1. However, other authors use the Support Vector Machine to detect and downweight the impact of outliers on classification contrary to removing them. The following is a summary of how SVMs have been used by different authors to reduce the effect of outliers on effective parameter estimation.

[Yang et al. \[2007\]](#) proposes using a Weighted Support Vector (WSVM) to reduce the impact that outliers (support vectors) have on the classifier. Their proposed method assigns weights to the data points allowing the decision boundary to learn from the training data points depending on how important they are in the set (judging by the weight they have). The results of their work show that this model reduces the effect of outliers on the model's classification accuracies and hence an improvement to a standard SVM where outliers are present in the dataset during training.

In [Herbrich and Weston \[1999\]](#), their solution to dealing with outliers was to use an Adaptive Margin Support Vector Machine (AMSVM) which uses adaptive margins for each training point as opposed to an equivalent margin for all training instances. A bounding generalization error rate for this AMSVM is used to reduce their sensitivity to the outliers.

[Lin and Wang \[2002\]](#) make use of a Fuzzy Support Vector Machine (FSVM) where a fuzzy membership is used for each data point. Different observations make different contributions to the decision boundary during training, which tunes the classifier and observes its sensitivity to the support vectors (reducing impact of outliers on the classifier).

[Yang et al. \[2007\]](#), [Herbrich and Weston \[1999\]](#), and [Lin and Wang \[2002\]](#) show how SVMs have been used to reduce the impact of outliers (support vectors) on parameter estimation. However, using the SVM to relabel outliers as a data level approach to minimising the impact of outliers on modelling, has not been done before. This is the essence of this work, to use the Support Vector Machine to detect outliers and relabel them by using their distances away from the separating hyperplanes. To measure the effectiveness of this relabelling technique, class labels are removed and a cluster analysis is performed. To see how well the clustering picks out inherent classes, we then measure the degree of separability between classes making use of Fisher's Linear Discriminant Analysis (FLDA) evaluated using the Adjusted Rand Index (ARI) values. We then compare the ARI values of the relabelled dataset to when the outliers are removed following the same process used to relabel.

2.2 Clustering

Lior and Maimon [2005], Grün [2018] describe clustering as an unsupervised learning technique to determine new categories. It varies from classification in that classification is predictive in nature deriving patterns from the data and using the learnt patterns to predict for new observations as seen in James et al. [2013] and Kotsiantis et al. [2007]. Contrary to the predictive nature of classification, Grün [2018] describes clustering as having different goals. Some of the goals highlighted include as an exploratory mechanism to determine patterns in multivariate datasets. Another aim may be to divulge groups that match the unobserved heterogeneity in datasets. In this work, clustering is used with the primary goal to dredge up homogeneous groups such that the different groups are as distinct as possible and similar groups are as close in pattern as possible.

2.2.1 KMeans Clustering

There are various clustering algorithms including KMeans clustering, Density based clustering, Hierarchical clustering, probabilistic clustering to name a few. However in this work, KMeans clustering and the hierarchical clustering are used as will be seen in the latter Chapter 4. According to Nathiya et al. [2010] KMeans clustering makes use of the Expectation-Maximization algorithm to calculate the KMeans. The E-step assigns the data points to the closest cluster whereas the M-step calculates the centroid for each cluster. With a KMeans clustering algorithm, the goal is to assign data points to clusters with the nearest centroids where the centroid is the arithmetic mean of the observations in the cluster.

Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of data points, and K be the number of clusters, then the following steps are performed in a KMeans clustering algorithm, see Maniriho and Effendi [2018];

1. Determine the number of clusters
2. Select at random, K data observations as the initial cluster centroids
3. Calculate the distance between each data point $\mathbf{p}$ and the cluster centroid $\mathbf{q}$. The distance computed is Euclidean and given by;

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (2.13)$$

where $\mathbf{p} = (p_1, p_2)$ and $\mathbf{q} = (q_1, q_2)$

4. Assign each data point to the closest cluster centroid

$$\arg \min_{c_i \in C} d(c_i, x) \quad (2.14)$$

where c_i is the cluster centroid

5. Recalculate each cluster centroid

$$c_i = \frac{1}{S_i} \sum_{x_i \in S_i} x_i \quad (2.15)$$

where S_i is the points belonging to cluster i

6. If the centroid is still changing, repeat steps 3 to 5 until the cluster centroid converges

2.2.2 Hierarchical Clustering

With a hierarchical clustering algorithm, the goal is also to achieve high quality clusters evaluated in various ways such as cluster cohesion where cluster cohesion is the level of association between the observations of a cluster, see Kaur et al. [2015]. According to this author, since

the technique uses a graphical form that is a tree based diagram, quality may also be measured using measures such as silhouette index. Silhouette index measures similarity of observations in a cluster. Furthermore, the time elapsed may well be used, this is the time it took to complete the clustering.

2.3 Fisher's Linear Discriminant Analysis

The Discriminant Analysis is used to determine which variables discriminate between two or more groups. Fisher in this regard aims to transform the multivariate observations of x to univariate observations y obtained from populations π_1 and π_2 were as distinct as possible. According to [Okwonu and Othman \[2012\]](#) Fisher introduced Linear Discriminant Analysis(LDA) as a dimension reduction and classification approach which uses sample mean and covariance matrices. The authors [Okwonu and Othman \[2012\]](#) and [Welling \[2005\]](#) define Fisher's Linear Discriminant Analysis (FLDA) as a linear combination of observed variables that best separate between the known groups of the data observations.

[Okwonu and Othman \[2012\]](#) states that Fisher's classification rule maximizes the variation between samples variability to within samples variability. The objectives of FLDA are as follows;

- Maximize the distance between the means of the given groups
- Minimize the variation (called scatter) within each category

The purpose of Fisher's LDA otherwise termed Fisher's criterion is to maximize the following objective function on the transformation matrix W ;

$$J(W) = \frac{W^\top S_B W}{W^\top S_w W} \quad (2.16)$$

where S_B is the "between classes scatter matrix" and S_w is the "within classes scatter matrix" and the definitions of these scatter matrices are as follows;

$$S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^\top \quad (2.17)$$

$$S_w = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^\top \quad (2.18)$$

where $\bar{x}$ is the overall mean of the data observations, μ_c is the mean of the i th class and x_i is the i th sample of a specific class and $\top$ is the transpose of a matrix.

The problem of maximizing J is transformed into the constrained optimization problem;

$$\min_W - \frac{1}{2} W^\top S_B W \quad (2.19)$$

$$s.t. W^\top S_w W = 1 \quad (2.20)$$

Once the between class scatter matrices and within class scatter matrices have been calculated, a lower dimensional space is constructed as can be evidenced in [Tharwat et al. \[2017\]](#). The equation otherwise termed as Fisher's criterion is shown in the equation below;

$$\arg \max_W \frac{W^\top S_B W}{W^\top S_w W} \quad (2.21)$$

$$S_w W = \lambda S_B W \quad (2.22)$$

where λ represents the eigenvalues of the transformation matrix W where;

$$W = S_W^{-1} S_B \quad (2.23)$$

if S_W is non-singular

According to [Michie et al. \[1994\]](#), the two reasons for using FLDA it either;

- Maximizes the separability between classes in the form of least squares or
- Maximum likelihood

Each of the above approaches have been proven to arrive at the same conclusion however in this work the first approach was adopted.

The following is discussed in the next subsection; the commonly used metrics for clustering beginning with the Rand Index (RI) and the Adjusted Rand Index (ARI).

2.3.1 Rand Index (RI)

According to the author [Ndwapi \[2018\]](#), one way of assessing performance of classification algorithms is to use the classification error where there is need to know the true class labels and to compare them against the predicted class labels that result from the model. The Rand index is calculated as follows;

Let X = number of pairwise agreements, Y = number of pairwise disagreements , then

$$RI = \frac{X}{X + Y} \quad (2.24)$$

where $X + Y$ shows the total number of pairs. A pairwise agreement means that you assign the same label to two observations that belong to the same class and or you assign different labels to different observations belonging to different classes. The Rand Index can assume values ranging from 0 to 1 where 1 shows perfect class agreement and 0 the reverse.

The authors [Santos and Embrechts](#) , [Ndwapi \[2018\]](#) both agree that the RI does not adjust for chance making provision for the ARI.

2.3.2 Adjusted Rand Index (ARI)

[Santos and Embrechts](#) describes the ARI as a way of validating clustering since it measures the agreement between partitions. The ARI measures the correlation allying pairs of dataset elements not using information from class labels. Therefore because of this attribute about the ARI, it can be used to detect intricacies with classification algorithms especially in combination with common performance measures. [Yang \[2016\]](#) defines ARI in the following manner;

$$ARI(P^*, P) = \frac{\sum_{i,j} \binom{N_{ij}}{2} - \left[\sum_i \binom{N_i}{2} \sum_j \binom{N_j}{2} \right] / \binom{N}{2}}{\frac{1}{2} \left[\sum_i \binom{N_i}{2} + \sum_j \binom{N_j}{2} \right] - \left[\sum_i \binom{N_i}{2} \sum_j \binom{N_j}{2} \right] / \binom{N}{2}} \quad (2.25)$$

where N is the number of data observations, N_{ij} is the number data observations of the class label $C^*_j \in P^*$ allocated to the cluster C_i in the partition P , N_i is the number of data observations in cluster C_i and N_j is the of data observation in class C^*_j .

Lior and Maimon [2005] argue that common performance measures such as the Mean Squared Error(MSE), AUC (area in percentage under the Receiver Operating Characteristic (ROC) curve) compare the labels of the supervised classification algorithm with the target labels known prior. Therefore, this attribute about these performance measures imply that an evaluation on how the algorithm labelled the input data according to the required labels is being put to test by which if the output labels are switched, they may yield poor results.

According to Vrbik and Mcnicholas [2014] when assessing clustering algorithms, the Adjusted Rand Index (ARI) is used as can also be seen in Steinley [2004]. The underlying intuition here is that by having carried out classification, the classes are known and as such it becomes easier to measure the agreement partitions.

2.4 Mixture Models

In light of the models described above, and the limitations posed by each of the models especially the KMeans clustering, more robust models have been developed which make use of the underlying patterns within the datasets. This includes the use of such models as the mixture models where a mixture entails a group of different probability distributions to capture the different underlying patterns inherent in the data. With the KMeans clustering, the clusters followed a circular shape however, data does not always conform to this shape. This is where the mixtures such as the Gaussian Mixture Model comes into play to capture the elliptical shape that may be inherited by the data as opposed to the ellipsoidal shape rather captured by a KMeans clustering.

The *Gaussian Mixture Model* (GMM) extends on to the KMeans clustering algorithm catering for cluster overlaps in the data. That is to say, by means of soft assignment of data points as will be seen on the discussion on the GMM, data is not forced to conform to the set clusters but new underlying clusters may emerge for data that doesn't conform to the clusters that are already defined.

To fit a Finite Mixture Model, the *Expectation Maximization* (EM) algorithm is used which converges to a point known as the *Maximum Likelihood* (ML) which is an estimate of the parameters of the mixture, see Figueiredo and Jain [2002]. Below is a definition of a mixture, the algorithm followed towards getting the mixture parameters and an illustration of how the Maximum Likelihood function and the Log Likelihood functions converge. First, a definition of a mixture as seen in many publications as Redner and Walker [1984], Bouveyron [2014], Myung [2003], Andrews and McNicholas [2012] Ndwapu [2018], with a difference of notation at times notwithstanding the similarity of the general concept;

$$p(x|\Phi) = \sum_{i=1}^m \alpha_i p_i(x|\phi_i), x = (x_1, \dots, x_n)^T \in \mathbb{R}^n \quad (2.26)$$

where m is the number of components and the mixing proportions $\alpha_i > 0$ such that $\sum_{i=1}^m \alpha_i = 1$, $\Phi = (\alpha_1, \dots, \alpha_m, \phi_1, \dots, \phi_m)$ is the parameter vector and p_i is the density function for the i th component.

For a mixture of Gaussians, each component $p_i(x|\phi_i)$, is represented by a normal distribution of parameters $\phi_i = \{\mu_i, \Sigma_i\}$, which are the mean vector and the covariance matrix. The parameters of the Gaussian mixture are defined by the set $\Phi = (\alpha_1, \dots, \alpha_m, \phi_1, \dots, \phi_m)$

2.4.1 Parameter Estimation

Parameter estimation defines a phenomenon that estimates the parameters or constants of a given model from the observations used in that model, [Van den Bos \[2007\]](#), [Beck and Arnold \[1977\]](#).

2.4.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) defines a method used to find the density that is more likely to have generated the data. This method works by identifying the *Maximum Likelihood Function* which shows the likelihood that an observed sample is a function of possible parameter values. The likelihood functions that are to be maximised share the same property of concavity where a concave function is by definition a function which is twice differentiable, that is, having a second derivative that is strictly negative. Getting the second derivative of the function is to assert that the value obtained on the first derivative is in fact the maximum, see [Redner and Walker \[1984\]](#).

[Sugiyama \[2015\]](#) formally defines MLE as follows;

“A set of probability density functions specified by a finite number of parameters is called a parametric model. Let us denote a parametric model by $q(x; \theta)$, a parameter vector by θ , and the domain of parameters by Θ , respectively. Let b be the dimensionality of θ : ”

$$\theta = (\theta^{(1)}, \dots, \theta^{(b)})^T \quad (2.27)$$

where x is a random variable and θ is a parameter.

2.4.3 Maximum Likelihood Estimation for Gaussian Distribution

The mixture of distributions that will be used in this work is that of the normal distributions. This is because as more samples from a dataset, they tend to take the shape of a Gaussian or a normal distribution even if the underlying pattern in the data is not Gaussian. The attribute is defined as the *Central Limit Theorem* as seen in [Heyde \[2006\]](#). Below is a MLE for the parameters of the Normal distribution, but first, the definition of the probability density function.

Let $X = \{x_1, x_2, \dots, x_n\}$ given (μ, σ^2) where both μ and σ^2 are unknown, the pdf is given by;

$$f(x_i | \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right), x_i \in R \quad (2.28)$$

Then the likelihood function is given by;

$$\begin{aligned} L(\mu, \sigma^2) &= \prod_{i=1}^n f(x_i | \mu, \sigma^2) \\ &= \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \\ &= \prod_{i=1}^n (\sigma^2)^{-1/2} \prod_{i=1}^n (2\pi)^{-1/2} \prod_{i=1}^n \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \\ &= (\sigma^2)^{-n/2} (2\pi)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right) \end{aligned} \quad (2.29)$$

To manage the complexity of maximising the likelihood estimates, the log-likelihood function of the likelihood function itself is used instead and it must be noted that obtaining the parameter estimates by maximising the log likelihoods is the same as obtaining them by maximising the likelihood function except for it may prove to be quite complex to maximise the actual likelihoods given that we would be dealing with exponentials.

2.4.4 Log-Likelihood Function of the Gaussian distribution

The log likelihood function is given by

$$\begin{aligned}
 L(\mu, \sigma^2) &= \ln L(\mu, \sigma^2) \\
 &= \ln \left[(\sigma^2)^{-n/2} (2\pi)^{-n/2} \exp \left(\frac{-1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \right) \right] \\
 &= \ln(\sigma^2)^{-n/2} + \ln(2\pi)^{-n/2} + \ln \left(\exp \left(\frac{-1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \right) \right) \\
 &= -\frac{n}{2} \ln \sigma^2 - \frac{n}{2} \ln(2\pi) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2
 \end{aligned} \tag{2.30}$$

To get the parameter estimates, differentiate the log likelihood with respect to μ and σ^2 .

$$\frac{\partial L(\mu, \sigma^2)}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu) \tag{2.31}$$

Omitting the detailed work on these estimates, the resulting MLE's are given below for μ and σ^2 respectively;

$$\hat{\mu} = \bar{X} \tag{2.32} \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n} \tag{2.33}$$

2.4.5 Maximum Likelihood Estimation for the Gaussian Mixture Model

From Eq. (2.26), the parameters Φ of the Gaussian mixture model is given by

$$\Phi = (w_1, \dots, w_m, \mu_1, \dots, \mu_m, \Sigma_1, \dots, \Sigma_m) \tag{2.34}$$

Following the same process as the unimodal normal distribution, even though more complex now, the resulting parameter estimates obtained from equating the derivatives to zero the maximum likelihoods $\hat{w}_\ell$, $\hat{\mu}_\ell$, and $\hat{\Sigma}_\ell$ should satisfy;

$$\begin{aligned}
 \hat{w}_\ell &= \frac{1}{n} \sum_{i=1}^n \hat{n}_{i,\ell} \\
 \hat{\mu}_\ell &= \frac{\sum_{i=1}^n \hat{n}_{i,\ell} x_i}{\sum_{i'=1}^n \hat{n}_{i',\ell}} \\
 \hat{\Sigma}_\ell &= \frac{\sum_{i=1}^n \hat{n}_{i,\ell} (x_i - \hat{\mu}_\ell)(x_i - \hat{\mu}_\ell)^T}{\sum_{i'=1}^n \hat{n}_{i',\ell}}
 \end{aligned} \tag{2.35}$$

where $\hat{n}_{i,\ell}$ is called the responsibility of the ℓ th component for sample x_i .

The likelihood function shown below shows how complicated it becomes to analytically solve for the variables leading way for a technique which iteratively solves for these variables known as EM algorithm described in the following subsection.

$$\hat{n}_{i,\ell} = \frac{\hat{w}_\ell N(x_i; \hat{\mu}_\ell, \hat{\Sigma}_\ell)}{\sum_{\ell'=1}^m \hat{w}_{\ell'} N(x_i; \hat{\mu}_{\ell'}, \hat{\Sigma}_{\ell'})} \quad (2.36)$$

In the event that the Eq. (2.35) $\hat{n}_{i,\ell}$ is not be known the EM algorithm is used and it follows the following steps;

2.4.6 Expectation Maximization Algorithm

The EM algorithm is a widely used technique in the iterative computation of parameter estimates when the likelihood function is complicated in formation due to difficult maximization problems either to analyse or computationally or both, see [Gentle et al. \[2012\]](#), [McLachlan and Krishnan](#). The authors [Gentle et al. \[2012\]](#) further point out that this algorithm works on incomplete data problem and by filling the missing data. According to [Ndwapi \[2018\]](#), incomplete data refers to the unobserved class labels. However, if the problem is that of a complete data problem, the Maximum Likelihood estimation technique is used. Where it is difficult to differentiate the likelihood function, a technique known as the Newton Raphson is used to get the estimates but this technique will not be discussed here.

[McNicholas \[2016\]](#) defines the EM algorithm for Gaussian model based clustering to follow the steps;

- Initialization step; Set of the initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model
- Expectation step; Guess the value of $\hat{n}_{i,\ell}$ which are missing or incomplete data estimated using the observed data
- Maximization step; Substitute in the equations obtained in the MLE using the $\hat{n}_{i,\ell}$ that was obtained in the Expectation step. Use all the data to update the values of the parameters
- Convergence; Repeat the E-step and the M-step until it converges

Initialization Step

According to [McNicholas \[2016\]](#), the EM may be initialized in one of the following ways;

1. Random initialization

- Start EM with parameter values which provide the largest likelihood
- Repeat EM algorithm several times, each time start with a random initial position and ensure that the parameter values used provide the largest likelihood from the runs.
- Stop and start EM at random locations before it exhibits convergence using a large number of short runs of the EM

2. Recursive initialization

- Involves splitting randomly one of the G components into two components to get a $(G + 1)$ solution

3. Hierarchical initialization

Expectation step

The data X are presumed not complete and the complete data set $Y = (X, Z)$ is determined by estimating the set of variables $Z = \{z_1, z_2, \dots, z_M\}$ where each z_m is an N -dimensional vector $[z_m^1, z_m^2, \dots, z_m^N]$. The complete data Y has the following log likelihood;

$$\log p(Y|\Phi) = \sum_{l=1}^N \sum_{l=1}^M z_m^i \log[\alpha_i p(x^i|\phi_i)], \quad (2.37)$$

where,

$$z_m^i = P(m|x^i, \Phi^t) = \frac{\alpha_m^t p(x^i|\phi_m^t)}{\sum_{l=1}^M \alpha_l^t p(x^i|\phi_l^t)} \quad (2.38)$$

is the posterior probability and Φ^t is the parameter estimate obtained after t iterations.

Maximization step

Here, the parameters Φ^{t+1} are determined by the estimate of the variables z_m^i . For Gaussian mixture models this means re-estimating the α_m^{t+1} , the μ_m^{t+1} , and $\sum_{l=m}^{t+1}$ for each m as per the following;

$$\alpha_m^{t+1} = \frac{1}{N} \sum_{l=1}^N z_m^i, \quad (2.39) \quad \mu_m^{t+1} = \frac{\sum_{l=1}^N z_m^i x_i}{\sum_{l=1}^N z_m^i}, \quad (2.40)$$

and

$$\sum_{l=m}^{t+1} = \frac{\sum_{l=1}^N z_m^i (x_i - \mu_m^{t+1})(x_i - \mu_m^{t+1})^T}{\sum_{l=1}^N z_m^i} \quad (2.41)$$

Convergence and stopping the procedure

- The observed log-likelihood is increasing at each iteration, $l_0(\phi^{(t+1)}) \geq l_0(\phi^{(t)})$ with equality if and only if $Q(\phi^{(t+1)}|\phi^{(t)}) = Q(\phi^{(t)}|\phi^{(t)})$. This applies when Jensen's inequality is applied to the convex function $H(\phi|\phi^{(t)})$ and this is the *fundamental convergence property of the EM algorithm*.
- Lack of progress in the log-likelihood i.e when $l^{(t+1)} - l^{(t)} < \epsilon$ for all ϵ is small where $l^{(t)}$ is the observed log likelihood value from iteration t .

2.5 Relabelling Techniques

Relabelling and removing outliers as a data preprocessing task is a common procedure carried out to improve classification accuracies [Muhlenbach et al. \[2004\]](#), [Jiang et al. \[2020\]](#) and [Rebapragada and Brodley \[2007\]](#). In [Jiang et al. \[2020\]](#), techniques such as suppression of sensitive attributes, changing or relabelling class labels a task they termed as massaging the dataset. Here, a ranker (Naive Bayesian classifier) is used to select the data point which should be relabelled. Some of the tasks they take prior to modelling is sampling, carried out to reduce parameter estimates. In the work by [Sperrin et al. \[2010\]](#), a framework based on the EM algorithm is used,

probabilistic relabelling on Bayesian mixture models. In [Muhlenbach et al. \[2004\]](#), a preliminary filtering procedure based on the cut edge weight statistic is used. Here a geometrical neighborhood graph is used and a statistical test based on the cut edge weight for detecting the suspect points (outliers- those with higher cut edge weight statistic values). From this cut edge weight statistic, a training data is obtained and used as input to the learning algorithm known as 1-NN strategy. The results of this study show that removing outliers outperforms relabelling but as noise in the data increases, the generalization performance of the classifier decreases. It is evident from this study that that even though removing outliers outperforms relabelling, removal of outliers is only efficient to an extent bearing in mind that data is being lost and hence poor if not inaccurate or data reflective generalization rates.

2.6 Techniques used to measure distance of support vectors from the separating hyperplane

According to [Bernstein \[2009\]](#), norms are used to quantify vectors and matrices. Where a norm can be considered a mapping from a vector to a scalar used to estimate how big a vector is or how close one vector is from another. A norm can be considered as a generalization of the length of vectors and or matrices. According to [Aggarwal \[2003\]](#) in data mining applications choosing a distance function to use is not usually a trivial task. He states that even though most algorithms use Euclidean distance functions such as the L_2 -norm, they however have a limitation of dimensionality that is, usually limited to 2-3 dimensions. [Aggarwal et al. \[2001\]](#) further goes on to elaborate that the Manhattan distance metric (L_1 -norm) becomes more favourable than the Euclidean distance metric (L_2 -norm) in higher dimensional spaces. They further state that the L_k norm on the fractional distance function becomes more significant towards attaining better results. Norms have the following properties, see [Epperson \[2013\]](#) and [Bernstein \[2009\]](#);

Vector norms satisfy the following axioms:

$$\textit{Positivity} : \|x\| \geq 0, \text{ and } \|x\| = 0 \iff x = 0 \quad (2.42)$$

$$\textit{Homogeneity} : \|kx\| = |k|\|x\| \quad (2.43)$$

where k is a scalar in $\mathbb{R}$

$$\textit{Subadditivity} : \|x\| + \|y\| \leq \|x+y\| \quad (2.44)$$

where k is a scalar in $\mathbb{R}$, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$.

From the above stated properties the following vector norms are derived;

Let $x \in \mathbb{R}^n$, then

1. The 1-norm of x is defined by,

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (2.45)$$

2. The L_p norm is defined by,

$$\|x\|_p = \left\{ \sum_{i=1}^n |x_i|^p \right\}^{1/p} \quad (2.46)$$

3. In the case $p = 2$, then the L_2 norm is defined.

$$\|x\|_2 = \left\{ \sum_{i=1}^n |x_i|^2 \right\}^{1/2} \quad (2.47)$$

The L_2 norm is sometimes referred to as the Euclidean norm.

4. Finally, the L_∞ is defined by,

$$\|x\|_\infty = \max |x_i| \quad (2.48)$$

Matrix norms satisfy the following axioms:

$$\|Av\| \leq \|A\|\|v\| \quad (2.49)$$

$$\|AB\| = \max_{\|v\|=1} \|ABv\| \leq \max_{\|v\|=1} \|A\|\|Bv\| = \|A\|\|B\| \quad (2.50)$$

Let $X \in \mathbb{R}^{n \times n}$ be a data matrix, then

1. 1-norm of X is defined by

$$\|A\|_1 = \max_j \sum_i |A_{ij}| \quad (2.51)$$

2. ∞ -norm of X is defined by

$$\|A\|_\infty = \max_i \sum_j |A_{ij}| \quad (2.52)$$

3. The 2-norm of X is computed as follows;

$$\begin{aligned} \|A\|_2^2 &= \left(\max_{\|v\|_2=1} \|Av\| \right)^2 \\ &= \max_{v^T v=1} v^T A^T A v \end{aligned} \quad (2.53)$$

It is common practice to use L_2 norms to calculate the margins from the separating hyperplanes as well as the data points away from the margins. However, [Pedroso and Murata \[2001\]](#) do indicate through the exploitation of geometrical properties of various norms L_2 performs relatively similar to the two norms L_1 and L_∞ .

2.6.1 Mahalanobis distance

Mahalanobis distance is a technique used to measure distance of points from the class centroids, [McLachlan \[1999\]](#). Mahalanobis distance makes use of the class centres $\bar{x}$ and covariance matrix (C_x) while a new point x is categorized as belonging to either one of the classes. Mahalanobis distance is given using the following function.

$$M_{dist} = \sqrt{(x - \bar{x})C_x^{-1}(x - \bar{x})^\top} \quad (2.54)$$

While the Mahalanobis distance is usually used on its own to detect outliers, we use it side to side with the SVM which is another prominent outlier detection technique. The SVM is used to detect outliers along the margins and calculate their Mahalanobis distance from the class centroid. The combination of these two help us know exactly the points to consider for relabelling see Chapter [3](#).

2.6.2 Summary

The tools that are used throughout this thesis have been discussed in this Chapter. The SVM to detect outliers, distance measures to compute the distances of the support vectors from the hyperplanes, relabelling techniques to, Discriminant analysis techniques to measure the level of class distinctiveness. In the next Chapter is a detailed description and discussion of the methodology that was followed to achieve the set objectives.



3. Proposed Relabelling Technique

This Chapter gives a detailed description of the activities carried out to achieve the set objectives, in general the process method for this work. The first section, is an outline of the construction of the SVM. The resulting SVM is used to train data using different defined kernels to find the best fit for each dataset.

3.1 Constructing the SVM

Given the following decision boundary

$$w^\top x + b = 0, \quad (3.1)$$

where w is a data matrix, and b a constant, some data points must be on one side of the hyperplane and the others have to be on the other side. The data points must satisfy the constraints

$$w^\top x_i + b \geq 0 \quad \forall x \in \{1, 2, \dots, N\} \text{ where } y_i = 1, \quad (3.2)$$

and

$$w^\top x_i + b \leq 0 \quad \forall x \in \{1, 2, \dots, N\} \text{ where } y_i = -1, \quad (3.3)$$

Maximizing $d(w)$ is equivalent to minimizing w^2 so the constrained optimization problem in SVM is;

$$\begin{aligned} & \min \frac{1}{2} w^2 \\ & \text{subject to } y_i(w^\top x_i + b \geq 1) \quad \forall i \in \{1, 2, \dots, N\} \end{aligned} \quad (3.4)$$

Now, introduce the Lagrange Multipliers to get the optimal vales of w and b

$$\mathcal{L} = \frac{1}{2} w^2 - \sum_{i=1}^m \alpha_i (y_i (w^\top x_i + b - 1)). \quad (3.5)$$

Minimize Eq.(3.5) subject to w, b and α

$$\theta_D(\alpha) = \min_{w, b} \mathcal{L}(w, b, \alpha) \quad (3.6)$$

Get the partial derivatives;

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= w - \sum_{i=1}^m \alpha_i y_i x_i = 0 \\ w &= \sum_{i=1}^m \alpha_i y_i x_i \end{aligned} \quad (3.7)$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^m \alpha_i y_i = 0 \quad (3.8)$$

Given the partial derivatives Eq.(3.7) and Eq.(3.8), substitute w into $\mathcal{L}(w, b, \alpha)$ which on expansion and simplification gives;

$$D(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_j \alpha_i y_i y_j \quad (3.9)$$

The following dual problem is derived and fed to the optimizer in this case CVXOPT optimization package in Python.

$$\begin{aligned} & \max_{\alpha} D(\alpha) \\ & \text{such that } \alpha_i \geq 0 \\ & \sum_{i=1}^m \alpha_i y_i x_i \end{aligned} \quad (3.10)$$

In solving the dual problem Eq.(3.10), the deduced α values are substituted into Eq.(3.7). The support vectors are the values for which $\alpha_i > 0$ according to the Karush Kuhn Turker (KKT) conditions which are primal feasibility, dual feasibility and complementary slackness respectively as shown by Eq.(3.11), Eq.(3.12) and Eq.(3.13);

$$w = \sum_{i=1}^m \alpha_i y_i \Phi(x_i) \quad (3.11)$$

$$w = \sum_{i=1}^m \alpha_i y_i = 0 \quad (3.12)$$

$$\alpha_i [y_i (w^T \Phi(x_i) + b) - 1 + \xi_i] = 0 \quad 1, 2, \dots, N \quad (3.13)$$

3.2 Training the SVM model

The SVM was trained on different datasets using different training to testing set ratios. Training was done under the assumption that the classes may be unbalanced and that a model may perform dismally or exceptionally well as a coincidence of the training sample it is working with at a particular moment. This is a common problem in data mining referred to as random allocation. To cater for issues of random allocation and class imbalance, the following control measures were put in place;

1. **Random Allocation:** Data was resampled to obtain n combinations to take care of the situation where the obtained results may be a coincidence that the model learnt very well on the random training data chosen at that point or the model failed to generalize new data well. For optimum results, the mean average μ of the ARI scores was calculated as depicted in Table 4.2.
2. **Class Imbalance:** Here, there were two scenarios to observe;
 - Having majority of observations of a single class going towards training, leaving the model to test on completely new data it has not learnt from. To curb this effect, the data is stratified, that is equal percentages of the stratum were chosen for both training and for testing, say choose $n\%$ of each class to train with and the remaining $(100 - n)\%$ of each class to test with. This way, there is equal representation of each

class for both training and testing. This applies where the classes are balanced that is equal number of observations for each class in the data and is more of an imbalance created by random allocation.

- The case where the data is imbalanced that is the number of observations for a single class are not equal to the observations of the other class or classes of the same dataset. In this case, instead of just sampling $n\%$ of the data and setting it aside for training, select $n\%$ of each class in the dataset and combine these subsets to make up the training set and leave the remaining observations of each class for testing. That way, each class is well represented during training to enable the model to learn sufficiently from the data.

Training a model comes with the risk that for a sample chosen at random, it is possible for the model to perform dismally on that sample. This could be because of the sample that it happened to be working on. To combat this, the model is run several times to cater for these issues. One could suggest exhausting all the combinations of a given dataset but this is not computationally feasible as for even a small dataset as the Iris which has 150 observations, the combinations are very large. Alternatively one could run the model observing the error margin between the previous accuracy and the current accuracy. If on the next several iterations the error margin does not change, then a convergence is reached by which the iterations are retrieved. However, the question that remains is the possibility of exhausting the sampling combinations before achieving convergence by which it reverts back to exhausting the combinations. While it may be possible to converge in a reasonable number of iterations, another uncertainty is on the number of times no change is achieved on the error rate to conclude that the functions has converged. So for this work, 100 iterations were chosen but an optimal number of iterations to choose is beyond the scope of this work. This choice also gave samples for which the model best performed on giving values of 1.

The following algorithm indicates the process followed when training the SVM;

Algorithm 1: Training the SVM on different dataset scenarios

Input: Dataset scenario

Output: SVM accuracy scores for different kernels

```

1 kernels = [linear, polynomial degree 2, polynomial degree 3, radial]
2 bestAccuracy, currentAccuracy, averageAccuracy, = 0
3 accuracyList = []
4 for i ∈ kernels do
5   while training sample ≤ 100 do
6     1. Select n% of observations from the given dataset at random such that there is equal
7       representation for each class where n ∈ [60, 70, 80];
8     2. Train SVM on different training set ratios on the different scenarios;
9     3. Test the set;
10    4. Calculate current accuracy score for both the training and testing set;
11    5. Add currentAccuracy to accuracyList
12    6. Calculate averageAccuracy
13    7. Find the best accuracy
14   end
15   averageAccuracy, bestAccuracy
16 end

```

After the SVM model has been fit and the kernel that best fits a given dataset identified, the outliers identified as the support vectors are observed. The distance of the identified support vectors is calculated relative to their respective class centers comparing this distance to the other

class centers to come up with a criteria for relabelling the outliers. For each support vector in a given class, compute the euclidean distance of the support vector from the other class centroid with the intention of observing which centroid a point is closer to. Relabel the identified outliers looking at how far off they are from separating hyperplane as well as from the class centroid. In the next subsection, the relabelling process is elaborated.

3.3 Relabelling algorithm

Figure 3.1 is used to explain in more detail the relabelling process. While Figure 3.1 shows the position of the support vectors on a hard margin SVM Figure 3.2 shows how the support vectors are scattered in a soft margin SVM where some points are allowed to be on the other side of the margin over a soft penalty term in compromise for better robustness of the classifier;

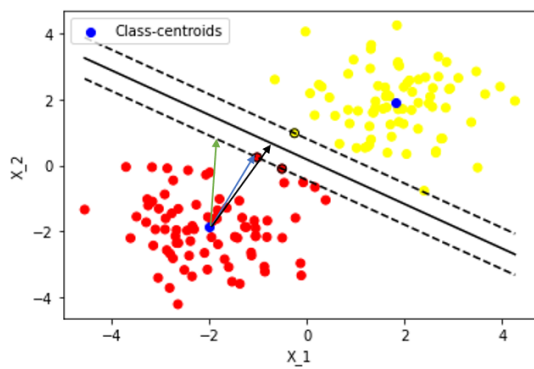


Figure 3.1: *Distance of the support vectors from the centroid and centroid from the separating hyperplane on a hard margin SVM*

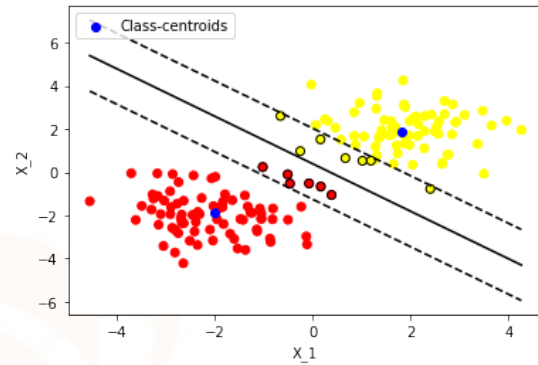


Figure 3.2: *Distance of the support vectors from the centroid and centroid from the separating hyperplane on a soft margin SVM*

1. Compute the distance of a class centroid (shown as a blue dot) from the separating hyperplane. This distance is illustrated using a black arrow.
2. Compute the distance (shown by the blue arrow) of a support vector which is circled in black, from the class centroid.
3. If the distance of a support vector from the class centroid (blue arrow) is greater than the distance of the separating hyperplane from the centroid, then the support vector in question belongs to the class it is located in. This is a clear case of a misclassification.
 - It should be noted that the orthogonal distance of the centroid from the separating hyperplane has to be obtained. For example, in the Figure 3.1, the green arrow is greater than the black arrow if extended to the hyperplane, but both are from the centroid to the hyperplane.
4. If the distance of the centroid from the separating hyperplane is greater than the distance of the support vector from the hyperplane, then the support vector has been correctly classified and should remain unchanged.

The activities for relabelling are summarised in the following algorithm.

Algorithm 2: Relabelling the support vectors**Input:** Dataset, supportVectorsList**Output:** Relabelled, Removed dataset

```

1 centroidList = [centroid1, centroid2]
2 foreach  $i \in \text{centroidList}$  do
3   | Compute the distance  $d_1$  of a class centroid from the separating hyperplane.
4 end
5 foreach  $x_i \in \text{supportVectorsList}$  do
6   | Compute the distance  $d_2$  of a support vector from the class centroid.
7 end
8 if  $d_2 > d_1$  then
9   | relabel
10 end
11 else
12   | break
13 end

```

It should be noted that the same algorithm as in Algorithm 2, is used but a point is removed instead of being relabelled.

The above steps describe relabelling using the Euclidean distance. However more predominantly, the Mahalanobis distance which also identifies as an outlier detection technique for multivariate data is used. When using the Mahalanobis distance, the class centroids are computed by getting the means from all the columns representing the attributes of the data and the covariance structure of the dataset is used. This means it takes into account how correlated the class features are to one another giving a more robust representation of the data. This captures the inherent structure in the dataset as compared to the Euclidean distance which seem to generalize on the attributes of the dataset.

The Mahalanobis distance creates covariance ellipses around the dataset with the ellipsoids more compact about the centre and increasing spread far away from the dataset. While the Mahalanobis will compute the distance of each point from the class centroid, and will show points that are furthest from their centroid all around the class centers, using it alongside the SVM helps to know exactly which points to consider and relabel about the margins reducing the computation for going through all the samples. Figure 3.3 shows a combination of the two outlier detection techniques, the SVM and the Mahalanobis distance, and it should be noted that the contours were drawn for illustration purposes and were not drawn to accuracy of the covariance ellipses.

The Euclidean distance is circular in shape which suggests there is no to minimum correlation between the variables of the dataset, while the Mahalanobis distance is ellipsoidal. This helps capture whichever pattern is in the data whether there is positive or a negative correlation in the data variables. Once the Mahalanobis distance of the support vectors is computed, steps of the relabelling algorithm are carried out with this distance measure techniques and the same comparisons for relabelling carried out. After the datasets have gone through the process for relabelling, measure how well the model will discriminate the datasets as to how they have been treated prior using the SVM.

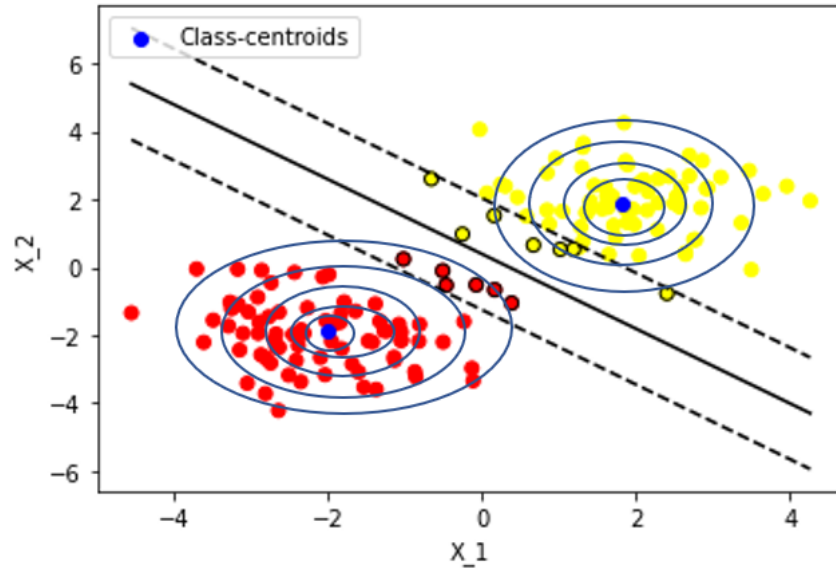


Figure 3.3: Positions of the support vectors from the separating hyperplane and as shown by Mahalanobis distance

3.4 Step 1: LDA on original datasets

- Select a percentage n of observations from the original dataset such that there is equal representation for each class i.e. n of each class for training and the remaining for testing where $n \in [60, 70, 80]$. This task was to take care of the random allocation issue where a model may have been assumed to perform well as a coincidence of the training set that it happens to be executing on at a given time, that is, process takes care of an incident where the model can pick majority of a single class for training leaving the majority of another class for testing resulting in the model failing dismally because the model is trained on one class and tested on another class.
- Perform LDA where training sets are $n\%$ of the observations in the original dataset then test the set. This process should be repeated at least k times and the average recorded for each ratio. This takes care of the randomness of train test selection.
- Calculate ARI for all n test sets in case most of the bad points went to the test set. Remove the labels to check comparing the labels of the original class against the predicted classes.

3.5 Step 2: LDA modified by the support vectors

- Split the dataset into training and test set (without the classes/labels) and apply the SVM to the complete original data and relabel.
- Select samples from the original data.
- Perform LDA on the transformed data (Drop the support vectors and perform LDA on k the randomly selected training sets).
- Compare with the original data the classification accuracy score that is, the ARI values. Calculate ARI based on original classes as opposed to the predicted classes with the SVM transformation.
- Observe how the models perform when values are missing.

The following algorithm is a summary of the activities carried out on the datasets when training a LDA model.

Algorithm 3: Assessing performance of the LDA on removed outliers compared to relabelled outliers

Input: Dataset scenario

Output: LDA values for k samples at a given training set ratio where k is the number of iterations

```

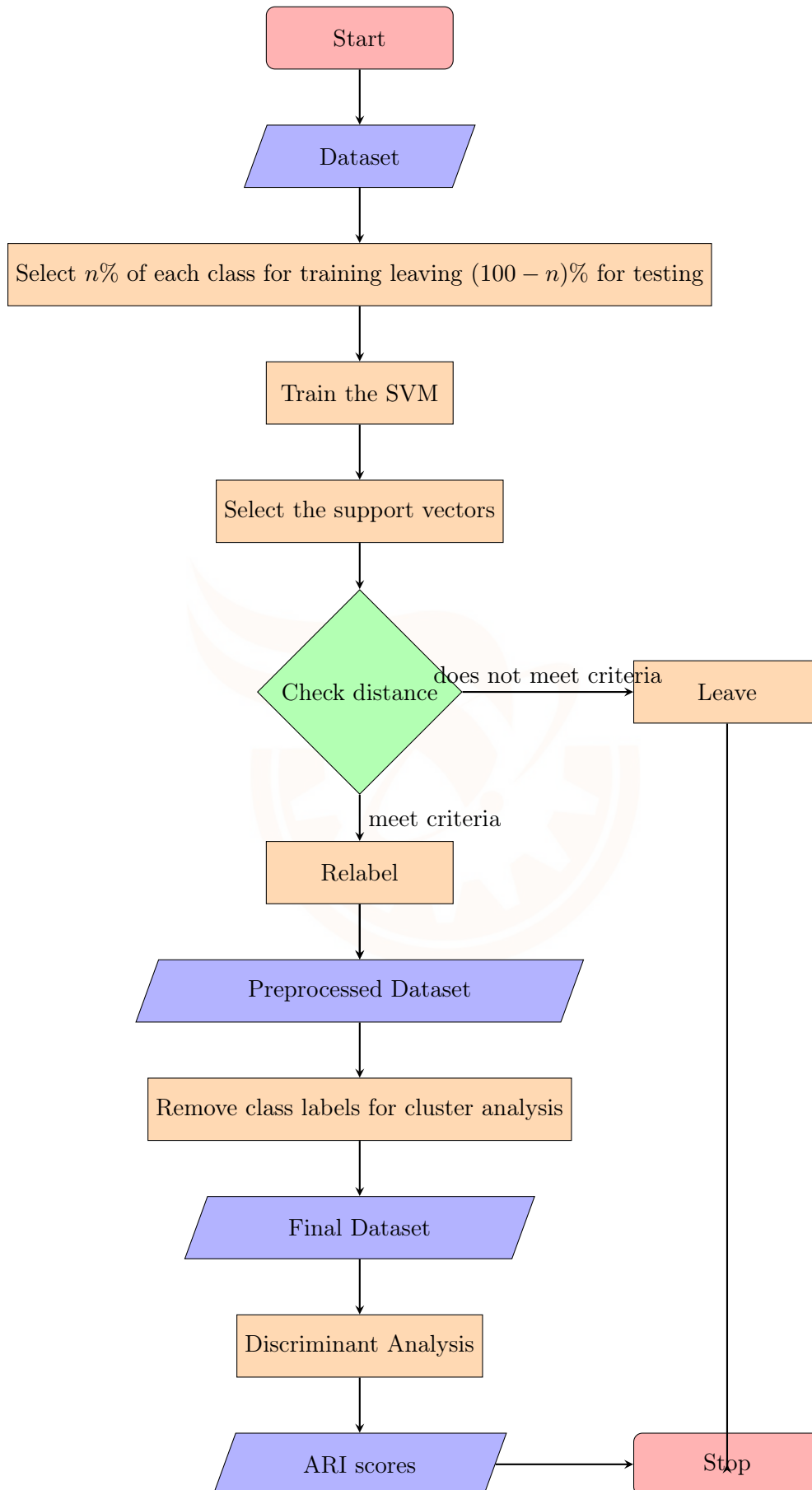
1 scenarios = [original, relabelled, removed]
2 bestARI, currentARI, average ARI, comparison, iteration = 0
3 ARList = []
4 for  $i \in \text{scenarios}$  do
5   while training sample  $\leq 100$  do
6     1. Select  $n\%$  of observations from the given dataset at random such that there is equal
7       representation for each class where  $n \in [60, 70, 80]$ ;
8     2. Perform LDA where training sets are  $n\%$  of the observations in the original dataset;
9     3. Test the set;
10    4. Calculate current ARI score for both the training and test set;
11    5. Remove the labels;
12    6. Compare the labels of the original class against the predicted classes;
13    7. Add currentARI to ARList
14    8. Calculate the averageARI
15    9. Find the bestARI
16   end
17   bestARI, average ARI
18 end

```

3.6 Gaussian Mixture Model for the dataset scenarios

The Gaussian Mixture models for the dataset scenarios are derived to map out the underlying inherent patterns. This means the GMM for an original dataset as well as its relabelled and removed outliers extraction are determined and modelled using LDA to see how well classification has been carried out.

The expectation is that by using the Support Vector Machine to detect outliers and relabel them, the impact that the support vectors will have on classification is being reduced. Data points that find themselves in opposite classes are assigned to those classes and those that find themselves on the margins are also assigned to classes they are most close to. In the latter parts of this work, the Linear Discriminant Analysis is incorporated to see how it performs on the dataset that has been preprocessed using the SVM as opposed to the original dataset before it was modelled with the SVM. An evaluation on how the LDA performs against the datasets on all scenarios and results are discussed in Chapter 4, Table 4.2. Figure 3.4 is a flow chart showing a summary of the process flow of the above defined activities for meeting the set objectives.

Figure 3.4: *Process flowchart*

3.7 Summary

In this Chapter, outliers detected and relabelled or removed depending on the scenario that needs to be achieved. For the different scenarios, further model using LDA to see the level of discrimination. The mixtures for the scenarios are also derived and modelled using LDA. The results for all these models are discussed in the next Chapter.



4. Results and Discussions

This Chapter is a presentation of the results obtained when modelling three datasets. First is a description of the datasets and their visual representations using scatter plots. In the next sections, the results are presented and an analysis of data is given and interpretation on how the models performed. The datasets are applied with the SVM to detect outliers otherwise shown to be the support vectors. The resulting datasets are passed on to different classification and discrimination models to observe the performance as seen in the Algorithm 1 in Chapter 3. The algorithms used are Fisher's Linear Discriminant Analysis (FLDA) evaluated using the Adjusted Rand Index (ARI). FLDA is used both on the datasets as they are and on the Gaussian Mixture Model form of the datasets.

4.1 Datasets

Three datasets Iris, Breast Cancer and Wine datasets were used for modelling. These datasets are multivariate ranging from 4 features for the Iris dataset, 14 for the Wine and 31 for the Breast cancer. In this work, the concern is on outliers in multivariate problems because multivariate data has multiple attributes which are used for modelling which means that more descriptions about the data are captured hence better accuracies during modelling.

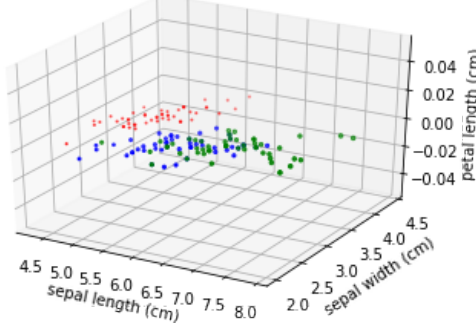
The datasets are visualized as seen in Figures 4.1, 4.4, and 4.6. These datasets are plotted using 3 random features of each respective dataset in 3D scatter plots, which is the highest dimension that could be visually captured. It is otherwise computationally expensive and difficult to visualize the data in such higher dimensions as displayed by these datasets. The data spread in the diagrams shows that the classes are overlapping meaning that there are outliers present in the data. To further show the overlap in some of the datasets, 2D scatter plots were plotted on 2 random attributes to further elaborate how one class may find itself plotted in the other class see Figure 4.3 for an elaboration of support vectors captured after applying the SVM to the Iris dataset.

4.1.1 Iris

The Iris dataset is a data matrix X of 150 by 4 where the features are evaluated into classes; setosa, versicolor and virginica having the class labels 0, 1 and 2 respectively. The Iris dataset was originally presented by Fisher [1936]. The columns are abbreviated as sl, sw, pl, pw for sepal length, sepal width, petal length and petal width respectively. This dataset has distinct classes which have to be separated following the similarity of the class attributes meaning that a class contains observations that have similar attributes or are very close in measurement. This is the essence of the problem of classification, proper assignment of observations or data points to the correct classes.

In the Figure 4.1 is a 3D scatter plot for the Iris dataset plotted on the first three attributes of the dataset before training the SVM model. In this figure, setosa is represented in red, versicolor in blue and virginica in green. While setosa tends to stand out, it is evident that versicolor and virginica tend to overlap implying that there are outliers. Figure 4.2 shows how the hyperplane separates between two classes shown in colors blue and red.

Scatter plot for the iris dataset before applying the SVM

Figure 4.1: The scatter plot for the classes *setosa*, *versicolor* and *virginica*

3D plot showing the separating hyperplane after applying the SVM

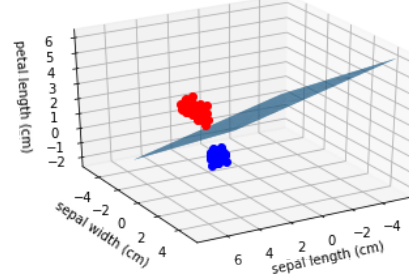
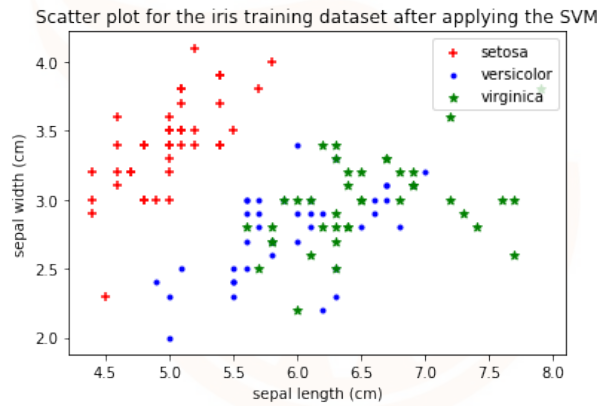
Figure 4.2: Hyperplane separating between *setosa* and *virginica*

Figure 4.3 shows a 2D scatter plot after the SVM is applied. The figure shows an overlap in the classes where data points belonging to the versicolor class plotted in blue find themselves classified as the setosa class plotted in red. Similarly, setosa data points have been classified as versicolor. There is also evidence of the classes versicolor and virginica classified as one another. When points are overlapping in this manner, they are identified to misclassified.

Figure 4.3: The scatter plot for the classes *setosa*, *versicolor* and *virginica* on the attributes, *sepal length* and *sepal width*

The SVM was trained on Iris dataset on the following training to test set ratios; 80:20, 70:30 and 60:40. Training was done on linear kernel, polynomial kernel and the radial kernel yielding performance scores of 1 across the trained SVM kernels, see Table 4.1 in Section 4.2. Upon being trained, two datasets were extracted, one with the outliers being relabelled and another, with the outliers being removed. The extracted datasets were further modelled using the LDA where the removed outliers dataset trained at 80% best performed in comparison to the other models. However, the other models were close in range, the relabelled dataset performed better than the control or the original dataset as shown in Table 4.2.

After training the SVM, LDA was used to observe how it performs, first on the original dataset for which no preprocessing had been carried out, then on a dataset for which outliers had been removed as opposed to when outliers are relabelled see Algorithm 1 in Chapter 3. The results are displayed in Table 4.2 see Section 4.2 and they show that LDA best performed on removed outliers. This could be because the outliers detected may not be enough to cripple the performance of the model or the generalization strength of the model on the dataset or that the model has been overfit.

On the issue of performance on the LDA, observation is made on how the model performed when the outliers had been relabelled using Euclidean distance as well as the Mahalanobis distance. There was also an investigation on how the model performed when the outliers treated using the proposed criteria where removed, see Table 4.2. When the outliers are treated using Euclidean distance, the performance scores obtained, are lower than on the original dataset. This can be attributed to the limitations of Euclidean on high dimensional sets. The Euclidean has limitations as follows see Ghorbani [2019]:

- Does not consider the correlation between closely correlated attributes
- Equal weight is assigned to class attributes implying that other attributes get more weight than they are worth.

A summary of results on the Iris dataset show that on LDA, Iris ARI score improved when the data was relabelled using Mahalanobis distance from the original dataset. There is an improvement from 0.8878 on the best performing training ratio to 1 on the relabelled dataset. A similar improvement on the removed outliers dataset is noticeable and this is expected because removing outlying points reduces the scatter but it comes at a cost of data loss. There is also improvement a Mixture of Gaussians on this dataset are used on the scenarios. Scores of 1 in general in the relabelled dataset as well as in the removed outliers dataset are observed.

4.1.2 Breast Cancer

This dataset was first presented by Wolberg et al. [1992] and is available in the Scikit Learn library Buitinck et al. [2013] in Python. The Wisconsin Breast Cancer dataset is a categorical data matrix X of 569 by 30. It has classes, benign and malignant encoded into numerical values 0 and 1 respectively where benign means the cell is non-cancerous and malignant means the cell is cancerous. The classes are unbalanced and distributed as 357 for the (benign) class and 212 for the (malignant) class.

The following Figure 4.4 shows the 3D scatter plot for the breast cancer dataset before applying the SVM. The spread in the plot shows that the classes are overlapping and that some classes find themselves well into the other class as can be seen traces of the blue class in the red class and vice versa. This means that the observations are misclassified that is assigned to the wrong class. Again here, the issue of data point removal becomes pertinent to say how many data points could be removed without negatively affecting modelling by way of giving performance scores that do not reflect the actual data.

Scatter plot for the breast cancer dataset before applying the SVM

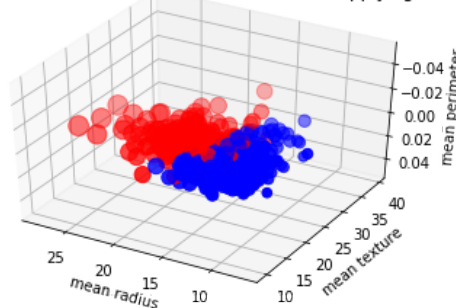


Figure 4.4: The scatter plot for the first three attributes mean radius, mean texture and mean perimeter

The following Figure 4.5 shows a 2D scatter on random features of this dataset to show the presence of outliers and class overlaps in the data.

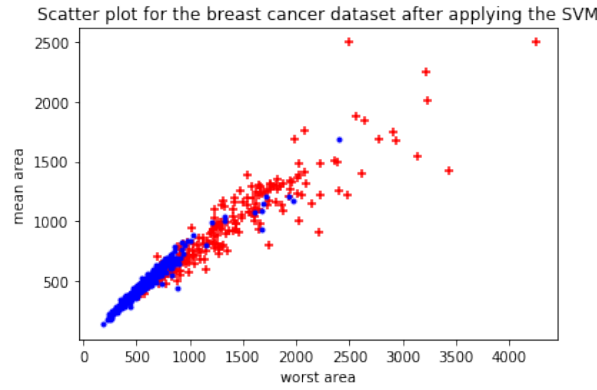


Figure 4.5: *The scatter plot for the first two attributes worst area and mean area*

The results show that the best performing kernel for this dataset was the Linear kernel trained at 70% and this was the model used to relabel the support vectors. On LDA, there is a slight decrease in the performance scores from 0.8625 in the Original dataset to 0.7155 in the relabelled dataset on the best performing models for these scenarios. This is also the phenomenon depicted with the GMM for this dataset.

4.1.3 Wine

Wine dataset was first presented by [Forina et al. \[1998\]](#) although the version used in this work is the preloaded version found on the Scikit Learn package in Python. The Wine dataset observes 13 categories for 3 different types of wine per the cultivars in a region found in Italy. The data matrix X measures 178 by 13 with 3 sorted labels 0, 1 and 2 respectively. The dataset is however unbalanced having 59 instances for the first class, 71 for the second class and 48 entries for the third class.

Figures 4.6 and 4.7 show the 3D and 2D scatter plots for the wine dataset before applying the SVM showing the overlaps in the classes. In each class, there are traces of the other two classes present in a single class. For example, if the red class is picked, there are traces of both blue and green observations sitting amongst the red class see more clearly on Figure 4.7. The same applies to the other two classes, where the green class has traces of both the red and the blue class and the blue class is as well riddled with both the green and the red class. Using the concept of the SVM explained earlier, this tells us that there are outliers in the dataset. When the SVM is applied, these are the data points identified as the support vectors.

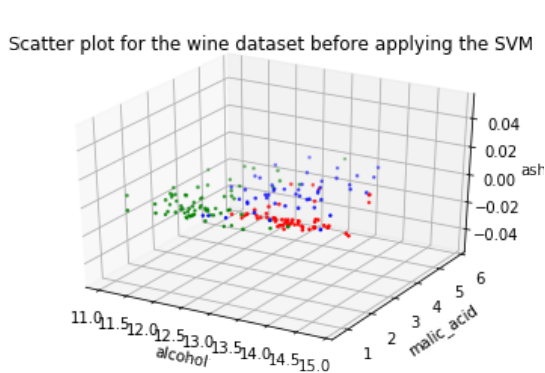


Figure 4.6: *The 3D scatter plot for the first three attributes alcohol, malic acid and ash*

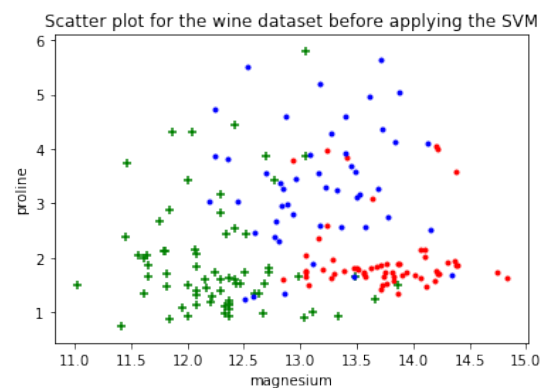


Figure 4.7: *The 2D scatter plot for the first two attributes alcohol, malic acid and ash*

Wine dataset well on the Radial kernel trained at 80%. Results for this dataset on the LDA shown in Table 4.2 showed a perfect classification for both the scenarios relabelled and removed datasets. A closer look into the relabelled dataset shows that against the control which is the original dataset, LDA showed an improvement from 0.9785 on the Original dataset to 1 on the relabelled dataset. In comparison to the removed dataset, for this dataset, the relabelled dataset performed as well as the removed dataset. An improvement on the GMM is also noticed.

Scatter plot for the Swiss banknote dataset before applying the SVM

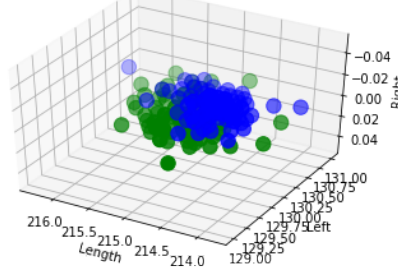


Figure 4.8: The scatter plot for the first two attributes length, left and right edges

Scatter plot for the Swiss banknote dataset before applying the SVM

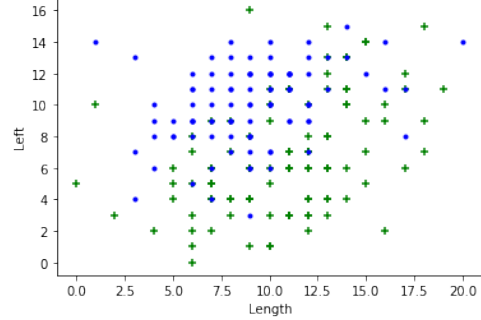


Figure 4.9: The scatter plot for the first two attributes length of the bill and left edge

On all the datasets discussed above, the data are multivariate so as to improve modelling and to facilitate the classification task. However, data misclassifications are evident in all sets contributing towards the classification problem. In all datasets, outliers identified as the support vectors, contribute towards the data that may already be insufficient as of the case of Iris and Wine datasets which have a relatively smaller number of observations compared to the Breast Cancer dataset. This is to say, this work does not encourage removing data points as it is bad for modelling. The following section, show how relabelling outliers yields as much competitive results as removing outliers in support of not easily losing data points.

4.2 Discussion

This section is a summary of the results obtained and discussions on them. First, the performance of the SVM on different kernels to show that there was close margin across all the datasets. The SVM was trained on different training to test set ratios, see results on Table 4.1, and the best performing kernel was chosen based on which kernel and which ratio was the best. What is also discussed in this section is how the 3 dataset scenarios; original, relabelled and removed dataset performed on the LDA where each dataset has these three scenarios for observation, see Table 4.2. These dataset scenarios are described more closely as;

- Original dataset; The unprocessed dataset by means of not being applied with the SVM prior. Note that in some datasets, necessary preprocessing such as normalizing the data or encoding labels had to be done without losing the context of this work i.e relabelling or removing data points.
- Relabelled dataset; The dataset extracted upon being applied with the SVM and the outlying data points relabelled using the developed relabelling technique.
- Removed dataset; The dataset extracted upon being applied with the SVM and the outlying data points removed using the developed relabelling technique.

More results and discussions follow on how the models were implemented, the issues that were encountered during modelling and how they were addressed.

Dataset	Train/ Test set	Linear kernel (%)			Polynomial Degree 2			Polynomial Degree 3 (%)			Radial (%)		
		80	70	60	80	70	60	80	70	60	80	70	60
Iris	Train	1	1	1	1	1	1	1	1	1	1	1	1
	Test	1	1	1	1	1	1	1	1	1	1	1	1
Breast Cancer	Train	0.965	0.9604	0.9607	0.8285	0.8898	0.9319	0.838	0.9019	0.9314	0.8339	0.8323	0.9327
	Test	0.9462	0.9478	0.9414	0.8098	0.8662	0.8995	0.8219	0.8802	0.9029	0.8124	0.8126	0.9019
Wine	Train	0.9877	0.9893	0.964	0.9998	0.9998	0.9995	0.9999	0.9997	1	0.9994	0.9989	0.9994
	Test	0.9769	0.9722	0.9653	0.9706	0.9724	0.9688	0.9743	0.9734	0.9664	0.9777	0.9706	0.9655

Table 4.1: Table showing the performance of the SVM on different kernels at different training to test set ratios

Dataset	Original			Distance measure	Relabelled			Removed		
	80	70	60		80	70	60	80	70	60
Iris	0.875	0.8843	0.8878	Euclidean	0.6676	0.6703	0.6631	1	1	1
				Mahalanobis	1	1	1	1	1	1
Breast Cancer	0.8613	0.8625	0.8533	Euclidean	0.5986	0.5895	0.5915	0.876	0.8684	0.8772
				Mahalanobis	0.7155	0.7055	0.7137	0.8661	0.8721	0.868
Wine	0.9785	0.9736	0.9776	Euclidean	1	1	1	1	1	1
				Mahalanobis	1	1	1	1	1	1

Table 4.2: Table showing the mean ARI scores performed on different datasets at different training to test set ratios

Dataset	Original			Distance measure	Relabelled			Removed		
	80	70	60		80	70	60	80	70	60
Iris	0.8772	0.8803	0.8852	Euclidean	0.6941	0.6891	0.6521	1	1	1
				Mahalanobis	1	1	1	1	1	1
Breast Cancer	0.8493	0.8555	0.853	Euclidean	0.6027	0.5873	0.6027	0.868	0.8736	0.8788
				Mahalanobis	0.7119	0.7112	0.7207	0.8819	0.8816	0.8776
Wine	0.7796	0.7968	0.7917	Euclidean	0.8642	0.8537	0.8556	1	0.9989	1
				Mahalanobis	1	1	1	1	1	1

Table 4.3: Table showing LDA results for the GMM performed on datasets

4.2.1 Summary of the performance of the SVM on various datasets

Choice of kernel

Table 4.1 shows how the kernel used for modelling was chosen. Different SVM kernels were trained on the three experiment datasets; Iris, Breast Cancer and Wine using different training to test set ratio, 80:20, 70:30 and 60:40. The SVM accuracy scores that were of more interest were the result of how well the SVM model predicted the labels in the test set having learned the patterns from the training set depending on which ratio was being used at the moment. It should however be noted that the table shows the accuracy scores on both the training set and the testing set in order to make reference to the training performance of the model when testing.

The 90% percent ratio was not considered under the assumption that the resulting models could be overfit that is learning patterns on the training set way too well and poorly generalizing on other data. On training using the polynomial kernels, as far as Degree 3 kernels were trained on the basis that, the higher dimensions you go, the more complex the computations are.

Results on training the SVM

From the Table 4.1, it shows that Iris depicted perfect classification giving scores of 1 across all the SVM kernels. For Breast Cancer dataset, the best performing kernel was the Linear kernel trained at 70% while it is also noticeable that the margins for this dataset were very close. For the Wine Dataset best performing kernel was the Radial kernel trained at 80%. It should also be emphasized that consideration was on the testing set performance. The model's accuracy scores are defined as the ratio of the predicted labels against the true labels, and was facilitated Scikit learn API see Buitinck et al. [2013] in Python. This means that the predicted labels must exactly match the true labels hence this metric.

4.2.2 Performance scores of the LDA on different datasets

After training the datasets using the simple SVM and identifying the outliers, the proposed relabelling technique was used to come up with two datasets, one where the outliers were relabelled and one where the outliers were removed. All the dataset scenarios, original, relabelled and removed datasets were taken up for further modelled using the Linear Discriminant Analysis. Table 4.2, shows the mean ARI values obtained from training the LDA. The developed relabelling technique was implemented using the Euclidean and it done alongside the Mahalanobis distance. From Table 4.2, the results show that where the Euclidean distance was used, the performance was rather lower compared to the Mahalanobis. This is due to the limitation of the Euclidean distance in multivariate data.

The results show an improvement on average from the original dataset to the relabelled dataset on treatment using the Mahalanobis distance. On Iris, the model improved from 0.8824 to 1 on average from the Original to the relabelled set. Breast Cancer on the other hand shows a decrement from 0.8590 to 0.7116 on average from the original to the relabelled dataset. The Wine dataset set showed an improvement from 0.9766 to 1 on average from the original to the

relabelled dataset as well. However removing outliers shows to yield the best results but this is a system that is discouraged on the basis of data loss and yielding models that are not data reflective.

4.2.3 Performance scores of the LDA on the GMM of the different datasets

The Gaussian Mixture Models of the various datasets were also extracted and LDA was run for the different scenarios. The results shown in Table 4.3 also show a general pattern similar to that discussed in subsection 4.2.2, where datasets that were relabelled using Mahalanobis distance perform well and in some cases yield ARI score of 1 for instance the Iris and the Wine dataset. Breast Cancer was the only exception in so far as obtaining perfect classification. The discrepancy could be worth investigating in future however the results are also arguably competitive ranging from 0.7112 on training at 70% to 0.7207% on training at 60%, on average being very close to 1.

4.3 Linear Discriminant Analysis on a Gaussian Mixture model on the datasets

Datasets Iris, Breast Cancer and Wine were modelled using mixtures of Gaussians and applied to an LDA. This was to improve on the KMeans clustering carried out prior. The Kmeans did not capture well the clusters and so mixtures of Gaussians were adopted to capture well the patterns underlying the data. The number of clusters were initialised using KMeans which uses the Elbow method to show the optimal number of clusters k . A sample graph depicting the elbow method on the Iris dataset is shown on Figure 4.10. The graph begins to converge when the number of clusters is 3 depicting the optimal number of clusters.

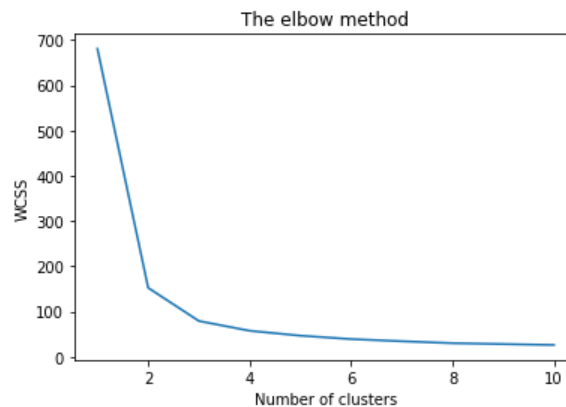


Figure 4.10: *Elbow method to determine the number of clusters for a KMeans clustering on the Iris dataset*

The LDA results shown on the Table 4.3 indicate a general improvement on the performance of the LDA when mixtures of Gaussians are absorbed than not. This shows that the mixture models capture better the underlying patterns in the data compared to when just modelling the data as it is, see also Table 4.4 for the scores obtained on modelling. On this experiment the Gaussian Mixtures were used for all the datasets discussed as well as the changes that were made on the data that is relabelled data and data for which the outliers had been removed.

Moving on to when relabelling and removing was done using the Mahalanobis distance, there is a noticeable improvement especially from the original set to relabelling in the Iris and Wine datasets. An increment from 0.8809 to 1 on average is seen on the Iris dataset and an increment

Dataset	Original	Relabelled	Removed
Iris	0.5033	0.4666	0.4889
Breast Cancer	0.5137	0.362	0.4784
Wine	0.3646	0.3152	0.3918

Table 4.4: Table showing the silhouette scores of the performance of KMeans clustering on GMM's

from 0.7894 to 1 on average on the wine dataset. However on Breast Cancer there is slight decrease from 0.8526 to 0.7146 in Breast Cancer dataset, however the results are relatively as good.

On the comparison of relabelled dataset compared to removed outliers dataset, note should well be taken that removing outliers outperforms relabelling and it is expected because since the SVM is primarily constructed from the support vectors, when they are removed, it means the class centroids are modified and the remaining data have more distance from the separation hyperplanes and the class compactness is better improved hence better model accuracies.

The performance of the GMM on LDA was measured using the silhouette score where a silhouette score is a measure ranging between $[-1,1]$ where 1 depicts a compact cluster which is distant from the other clusters. Values near 0 show an overlap of clusters meaning samples are closest to the decision boundary of other clusters. Negative scores depict a misclassification that is to say that observations may have been assigned to the wrong clusters.

The Table 4.4 shows that on average the clusters have been well separated judging by how far they are from 0. It is important to note that the ratio of training to testing did not have an impact on the silhouette scores. hence only the non-repetitive values were captured and depicted in this table, where non-repetitive means omitting the different ratios of training to testing.

It should be noted that in addition to Iris Breast Cancer and Wine dataset, Banknote Authentication and Swiss Banknotes were also considered. However the outliers identified for these datasets did not satisfy the developed relabelling criteria and hence the results not documented. In future, ways to improve the algorithm can be investigated to make it more rigorous and robust to identifying outliers as well as relabelling them.

5. Conclusions and Future work

The main objective of this work was to detect outliers and downweight their impact on modelling. The technique used to detect the outliers, used the concept of the Support Vector Machine (SVM) and the support vectors where the outlierness of the support vectors have been demonstrated with the use of the separating hyperplanes. In this work, the performance of data treated using the developed relabelling technique using two distance measure algorithms, Euclidean and Mahalanobis has been demonstrated. Euclidean as evidenced by literature has a generalization limitation when it comes to high dimensional data and evidently so, from the results in both Tables, 4.2 and 4.3. Mahalanobis with a stronger statistical power and relative to the dataset in place has shown to perform better than Euclidean distance and is also backed up by the literature on the detection of outliers in multivariate data.

Gaussian Mixture Models as well, have performed well in capturing data patterns unsupervised. Just as with LDA on the datasets, where the data was treated using the Mahalanobis distance relabelling technique, performance was better compared to on the Mixture of Gaussian in the original datasets. Gaussian Mixtures were meant to clearly depict the inherent data patterns in the data.

It is also worth mentioning some of the key issues that this research brought to light. This research has shown that how well a model performed on data may be a result of the sample being handled in a given time. This therefore, requires one to be methodical in how they handle this phenomenon such as addressing issues on the randomness of samples, optimal training and testing ratios as well as choosing the best kernel for a Support Vector Machine on a given dataset. The choice on the number of clusters in clustering and initialization of the EM-algorithm were also key questions that needed to be addressed and it has been shown that these are active research areas that are expected to keep on improving.

When you remove outliers, there is a cost to modelling that data is being lost, a phenomenon that is rather discouraged in data mining fields. Removing outliers yields better results and evidently so, the techniques and models used show that indeed there is bound to be better results given that the model may be underfit that is overgeneralizing on the given data. Arguably so, the models adopted in this work for instance, LDA, clustering and the GMM as seen in Chapter 2 in some sense or another work by moving the class centers such that in the end, the clusters or classes depending on whichever model is being used are compact and distinct. Therefore if points that tend to make it difficult to achieve this distinctness are removed it therefore means that the models will yield better results. In conclusion this research has shown that indeed other than removing outlying data points, relabelling them instead produces relatively good results that help discourage data point removal.

5.1 Future work

While working on this research, areas that could be investigated more have come to light and are penned out next.

- Model the identified support vectors done taking consideration the statistical information on the data, the class mean, the variance covariance structure standard deviation of the data.
- Work on algorithm to find an optimal relabelling criteria

-
- Observe the relationship between training accuracy and testing accuracy and the random effect
 - Find an optimal number of times that a model should be run to assert confidence that the model did not happen to perform extremely well on the data that it happened to be working on.



Bibliography

1. Edgar Acuña and Caroline Rodriguez. On detection of outliers and their effect in supervised classification. *University of Puerto Rico at Mayaguez*, 15, 2004.
2. Charu C Aggarwal. Towards systematic design of distance functions for data mining applications. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 9–18, 2003.
3. Charu C Aggarwal. *Outlier analysis second edition*. Cham: Springer, 2016.
4. Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.
5. Aida Ali, Siti Mariyam Shamsuddin, Anca L Ralescu, et al. Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl*, 7(3):176–204, 2015.
6. Jeffrey L Andrews and Paul D McNicholas. Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing*, 22(5): 1021–1029, 2012.
7. Ricardo Barandela, José Salvador Sánchez, V Garca, and Edgar Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.
8. James Vere Beck and Kenneth J Arnold. *Parameter estimation in engineering and science*. James Beck, 1977.
9. Dennis S Bernstein. *Matrix mathematics: theory, facts, and formulas*. Princeton university press, 2009.
10. Charles Bouveyron. Adaptive mixture discriminant analysis for supervised learning with unobserved classes. *Journal of classification*, 31(1):49–84, 2014.
11. Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
12. Gongxian Cheng. *Outlier management in intelligent data analysis*. PhD thesis, University of London, 2002.
13. Michiel Debruyne et al. An outlier map for support vector machine classification. *The Annals of Applied Statistics*, 3(4):1566–1580, 2009.
14. D Divya and Suvanam Sasidhar Babu. Methods to detect different types of outliers. In *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pages 23–28. IEEE, 2016.
15. James F Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.

16. H Jair Escalante. A comparison of outlier detection algorithms for machine learning. In *Proceedings of the International Conference on Communications in Computing*, pages 228–237, 2005.
17. Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.
18. Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
19. M Forina et al. An extendible package for data exploration, classification and correlation. institute of pharmaceutical and food analysis and technologies, 1998.
20. James E Gentle, Wolfgang K Härdle, and Yuichi Mori. Springer handbooks of computational statistics. 2012.
21. Hamid Ghorbani. Mahalanobis distance and its application for detecting multivariate outliers. *Facta Univ Ser Math Inform*, 34(3):583–95, 2019.
22. Bettina Grün. Model-based clustering. *Handbook of mixture analysis*, pages 163–198, 2018.
23. Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.
24. D Hawkins. *Identification of Outliers*. Springer Science & Business Media, 2013.
25. Ralf Herbrich and Jason Weston. Adaptive margin support vector machines for classification. 1999.
26. CC Heyde. Central limit theorem. *Encyclopedia of actuarial science*, 1, 2006.
27. Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.
28. Wenjie Hu, Yihua Liao, and V Rao Vemuri. Robust support vector machines for anomaly detection in computer security. In *ICMLA*, pages 168–174, 2003.
29. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
30. Jinfang Jiang, Guangjie Han, Lei Shu, Mohsen Guizani, et al. Outlier detection approaches based on machine learning in the internet-of-things. *IEEE Wireless Communications*, 27(3): 53–59, 2020.
31. Ferdinand Kaiser. Robust support vector machines for implicit outlier removal. Master’s thesis, 2013.
32. Bharati Kamble and Kanchan Doke. Outlier detection approaches in data mining. *International Research Journal of Engineering and Technology (IRJET)*, 4:634–638, 2017.
33. Puneet Jai Kaur et al. Cluster quality based performance evaluation of hierarchical clustering method. In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pages 649–653. IEEE, 2015.
34. Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006a.
35. Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006b.

36. Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.
37. Mark Last and Abraham Kandel. Automated detection of outliers in real-world data. In *Proceedings of the second international conference on intelligent technologies*, pages 292–301, 2001.
38. Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE transactions on neural networks*, 13(2):464–471, 2002.
39. Rokach Lior and Oded Maimon. Clustering methods. *Data mining and knowledge discovery handbook*, pages 321–352, 2005.
40. Harshada C Mandhare and SR Idate. A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 931–935. IEEE, 2017.
41. Pascal Maniriho and Ari Effendi. Examining the performance of k-means clustering algorithm. *Journal of Research in Engineering, Science and Management*, 1:1–5, 2018.
42. Geoffrey J McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.
43. Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*.
44. Paul D McNicholas. Model-based clustering. *Journal of Classification*, 33(3):331–373, 2016.
45. Donald Michie, David J Spiegelhalter, and Charles C Taylor. Machine learning, neural and statistical classification. 1994.
46. M Syed Mohamed and T Kavitha. Outlier detection using support vector machine in wireless sensor network real time data. *Int J Soft Comput Eng*, 1(2), 2011.
47. M Mohammadi and M Sarmad. Outlier detection for support vector machine using minimum covariance determinant estimator. *Journal of AI and Data Mining*, 7(2):299–309, 2019.
48. R Mollineda, R Alejo, and J Sotoca. The class imbalance problem in pattern classification and learning. In *II Congreso Espanol de Informática (CEDI 2007)*. ISBN, pages 978–84, 2007.
49. Fabrice Muhlenbach, Stéphane Lallich, and Djamel A Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, 2004.
50. In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.
51. G Nathiya, SC Punitha, and M Punithavalli. An analytical study on behavior of clusters using k means, em and k* means algorithm. *arXiv preprint arXiv:1004.1743*, 2010.
52. Nkumbuludzi Ndwapi. *Robust Discriminant Analysis With Asymmetric Classes*. PhD thesis, McMaster University, 2018.
53. Friday Zinzendoff Okwonu and Abdul Rahman Othman. A model classification technique for linear discriminant analysis for two groups. *International Journal of Computer Science Issues (IJCSI)*, 9(3):125, 2012.
54. Jason W Osborne and Amy Overbay. The power of outliers (and why researchers should always check for them). *Practical Assessment, Research, and Evaluation*, 9(1):6, 2004.

55. João Pedro Pedroso and Noboru Murata. Support vector machines with different norms: motivation, formulations and results. *Pattern recognition letters*, 22(12):1263–1272, 2001.
56. Umaa Rebbapragada and Carla E Brodley. Class noise mitigation through instance weighting. In *European conference on machine learning*, pages 708–715. Springer, 2007.
57. Richard A Redner and Homer F Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239, 1984.
58. Jorge M Santos and Mark Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In editor, editor, *International conference on artificial neural networks*.
59. Gaganpreet Sharma. Pros and cons of different sampling techniques. *International journal of applied research*, 3(7):749–752, 2017.
60. Michael R Smith and Tony Martinez. Improving classification accuracy by identifying and removing instances that should be misclassified. In *The 2011 International Joint Conference on Neural Networks*, pages 2690–2697. IEEE, 2011.
61. Matthew Sperrin, Thomas Jaki, and Ernst Wit. Probabilistic relabelling strategies for the label switching problem in bayesian mixture models. *Statistics and Computing*, 20(3):357–366, 2010.
62. Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.
63. Masashi Sugiyama. *Introduction to statistical machine learning*. Morgan Kaufmann, 2015.
64. Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.
65. Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-Pacific Web Conference*, pages 99–109. Springer, 2008.
66. Shuchita Upadhyaya and Karanjit Singh. Classification based outlier detection techniques. *International Journal of Computer Trends and Technology*, 3(2):294–298, 2012.
67. Adriaan Van den Bos. *Parameter estimation for scientists and engineers*. John Wiley & Sons, 2007.
68. Irene Vrbik and Paul D McNicholas. Parsimonious skew mixture models for model-based clustering and classification. *Computational Statistics & Data Analysis*, 71:196–210, 2014.
69. Max Welling. Fisher linear discriminant analysis. department of computer science, university of toronto. Technical report, Technical Report, 2005.
70. William H Wolberg, W Nick Street, and Olvi L Mangasarian. Breast cancer wisconsin (diagnostic) data set [uci machine learning repository], 1992.
71. Xiaodan Xu, Huawen Liu, Li Li, and Minghai Yao. A comparison of outlier detection techniques for high-dimensional data. *International Journal of Computational Intelligence Systems*, 11(1): 652–662, 2018.
72. Xulei Yang, Qing Song, and Yue Wang. A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):961–976, 2007.
73. Yun Yang. *Temporal data mining via unsupervised ensemble learning*. Elsevier, 2016.