



UNIVERSIDADE D
COIMBRA

Maria Leonor Inês Coelho

**THE JOINT-EFFECT OF IMBALANCED AND MISSING DATA:
A CHALLENGING TASK IN DATA ANALYSIS**

Dissertation in the context of the Master in Data Science and Engineering advised by
Professor Pedro Henriques Abreu and Miriam Seoane Santos and presented to the Faculty
of Sciences and Technology / Department of Informatics Engineering.

February, 2022

Faculty of Sciences and Technology
Department of Informatics Engineering

The joint-effect of imbalanced and missing data: a challenging task in data analysis

Maria Leonor Inês Coelho

Dissertation in the context of the Master in Data Science and Engineering advised by Professor Pedro Henriques Abreu and Miriam Seoane Santos and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

Coimbra, 2022



UNIVERSIDADE D
COIMBRA

This page is intentionally left blank.

*“If you can’t explain it simply,
you don’t understand it well enough.”*

- Albert Einstein

This page is intentionally left blank.

Abstract

The evolution of the technology increased exponentially the amount of available data and the complexity of it, which brought some data quality problems that affect negatively the performance of the data mining process. These data quality issues can be divided into two main categories: distribution-based, which includes class imbalance and small disjuncts, and feature-based, that includes missing data. These problems often occur together in real-world datasets, therefore, it is important to study how problems from one category affect issues from the other.

The interrelation among problems from the same category have already been studied while the relation between distribution and feature-based have yet to be researched. This thesis focus on this interrelation and how both problems affect the classification performance.

In this work, it is presented a study on some datasets characteristics and the effect they have on the imputation and classification performance. The considered characteristics were the size and number of features in a dataset, the Imbalance Ratio (IR), some complexity metrics and the distribution of the minority class. These characteristics do not have a high impact on the imputation performance while the IR and the distribution of the minority class highly affect the classification task. The higher the IR and the percentage of unsafe samples, the lower the performance will be. In conclusion, the classification will have worse results when a dataset has a higher complexity.

Keywords – missing data, imbalanced data, small disjuncts, data analysis

This page is intentionally left blank.

Resumo

A evolução da tecnologia aumentou exponencialmente a quantidade e complexidade dos dados, o que levou ao aparecimento de problemas ao nível dos dados que afetam negativamente o desempenho do processo de extração de conhecimento dos dados. Estes problemas podem ser divididos em duas categorias: problemas de distribuição, onde estão incluídos o não balanceamento dos dados e os *small disjuncts*, e de variáveis, onde se encontra o problema dos dados em falta.

A relação entre dificuldades da primeira categoria foi já estudada por alguns autores. No entanto, a relação entre problemas de cada uma das categorias ainda não foi abordada na literatura. Por isso, o foco desta tese é a inter-relação entre problemas de diferentes categorias e como é que esses problemas afetam o desempenho da classificação.

Neste trabalho, é apresentado um estudo sobre como algumas características de datasets afetam a imputação de dados em falta e a classificação dos dados. As características consideradas foram o tamanho e número de features num dataset, o IR, algumas métricas de complexidade e a distribuição da classe minoritária. Chegou-se à conclusão que estas características não têm um grande impacto na imputação mas, por outro lado, o IR e a distribuição da classe minoritária afetam bastante os algoritmos de classificação. Quanto menos balanceado um dataset é e mais dados *unsafe* tem, pior será o desempenho da classificação. Em conclusão, a classificação irá ter pior resultados em datasets com uma complexidade mais alta.

Palavras-chave – dados em falta, dados não balanceados, *small disjuncts*, análise de dados

This page is intentionally left blank.

Agradecimentos

Agradeço ao Professor Pedro Abreu pela ajuda incansável durante o último ano e pela confiança depositada em mim desde o início. Ajudou-me a crescer e ensinou-me imenso neste último ano. Deixo também um agradecimento à Miriam Santos por todos os conselhos e disponibilidade no decorrer deste trabalho.

Aos colegas e amigos que me acompanharam desde o primeiro dia da licenciatura, especialmente aos que comigo completaram o Mestrado em Engenharia e Ciência de Dados, agradeço todos risos e lágrimas, todas as almoçadas e noitadas, durante estes 5 anos

Um obrigada às minhas amigas de todas as horas, Beatriz, Mariana e Carolina, por ouvirem todos os meus lamentos e estarem comigo no fim de mais uma etapa da minha vida.

Por fim, o maior agradecimento vai para a minha família que me proporcionou os melhores anos até hoje. Sem o vosso apoio não estaria aqui. Um obrigado especial aos meus avós que me acolheram de braços abertos e me deram tanto apoio durante este tempo.

This page is intentionally left blank.

Contents

1	Introduction	1
1.1	Context and Motivation	2
1.2	Main Goals	4
1.3	Document Structure	4
2	State of the Art	7
2.1	Missing Data	7
2.1.1	Missing Data Mechanisms	8
2.1.2	Missing Data Imputation	9
2.2	Distribution-based irregularities	11
2.2.1	Class Imbalance	11
2.2.2	Small Disjuncts	14
2.3	Performance Evaluation Metrics	15
2.3.1	Imputation Quality	15
2.3.2	Imbalance Classification Quality	17
2.4	Datasets characteristics	19
2.4.1	Meta-features	20
2.4.2	Minority class distribution	21
2.5	Literature Review	23
2.5.1	Sequential Approaches	23
2.5.2	Interrelation between missing and imbalanced data	25
2.5.3	Summary	27
3	Experiments	30
3.1	Datasets	30
3.2	Preliminary Work and Assumptions	33
3.2.1	Missing Features and Class Imbalance	35
3.2.2	Missing Features and Complexity Metrics	38
3.2.3	Missing Features and Distribution Based	40
3.2.4	Main conclusions	41
3.3	Data irregularities and classification task	42

3.4 Final conclusions	50
4 Conclusion and Future Work	54
A Complexity Metrics Description	64
B Datasets Description	66
C Artificial Datasets Results	68
D Correlation for the Imputation Results	71
E Correlation for the Classification Results	75

Acronyms

ρ Pearson Correlation Coefficient. 16, 37, 38

R^2 Coefficient of Determination. 16

3Q Third Quartile. 39

AdaBoost Adaptive Boosting. 24, 28

ADASYN Adaptive Synthetic Sampling Method. 23, 26, 28

AUC Area Under the Receiver Operating Characteristic (ROC) Curve. 17, 19, 23, 25, 26, 28

AutoHPO Automated Hyperparameter Optimization. 23, 28

CART Classification and Regression Tree. 25, 28

CBOS Cluster Based Oversampling. 26

CBUS Cluster Based Undersampling. 26

CE Classification Error. 15

CGAIN Conditional Generative Adversarial Imputation Network. 26–28, 42

DNN Deep Neural Network. 23, 24, 28

DT Decision Tree. 23, 26, 28

EM Expectation-Maximization. 24, 28

ENN Wilson’s Edited Nearest Neighbor Rule. 13

FID Fuzzy-Based Information Decomposition. 25, 28, 42

FN False Negatives. 17, 24

FNR False Negative Rate. 23, 24, 28

FP False Positives. 17

- FPR** False Positive Rate. 19, 23, 24, 28
- G-Mean** Geometric Mean. 19, 23–25, 28
- GAIN** Generative Adversarial Imputation Network. 27, 28
- GM** Geometric Mean. 25, 28
- HEOM** Heterogeneous Euclidean Overlap Metric. 24
- IR** Imbalance Ratio. v, vii, xviii, 12, 13, 24, 27, 31, 35–38, 41, 42, 46, 48, 49, 51, 54, 55
- k-NN** k-Nearest Neighbours. xvii, xviii, 9, 10, 24, 25, 27, 28, 33–37, 41, 44, 69, 70
- KDD** Knowledge Discovery in Databases. 1
- LR** Logistic Regression. 26, 28
- MAE** Mean Absolute Error. 16, 34, 36, 39
- MAR** Missing At Random. 8
- MCAR** Missing Completely At Random. 8, 23, 24, 26, 27, 33, 34, 44
- MCC** Matthews Correlation Coefficient. 25, 28
- MI-MOTE** Multiple Imputation-Based Minority Oversampling Technique. xxi, 26, 28, 42, 44, 45, 47, 48
- MICE** Multiple Imputation by Chained Equations. xvii, xviii, 9, 10, 25–28, 33–37, 41, 69, 70
- Mix** Mixture Kernel-Based. 25
- ML** Machine Learning. 1, 2, 7, 19
- MNAR** Missing Not At Random. 8
- MR** Missing Rate. 34, 44
- MWM** Majority Weighed Minority. 26
- NB** Naive Bayes. 23, 28
- NN** Neural Network. 26, 28
- PCA** Principal Component Analysis. 23
- PPV** Positive Predictive Value. 18

- RF** Random Forest. xv, 10, 11, 23, 24, 26–28, 44, 46
- RFR** Random Forest (RF) Regression. 23, 24, 28
- RMSE** Root Mean Squared Error. 16, 27, 28, 34, 36, 37
- ROC** Receiver Operating Characteristic. xiii, 17, 19
- ROS** Random Oversampling. 26
- RUS** Random Undersampling. 25
- SMOTE** Synthetic Minority Oversampling Technique. xvii, 13, 14, 25–28
- SOMI** Self-Organizing Maps for Imputation. 25
- SVM** Support Vector Machines. 9, 23, 24, 28
- TN** True Negatives. 17
- TP** True Positives. 17
- TPR** True Positive Rate. 18, 19

This page is intentionally left blank.

List of Figures

1.1	Knowledge Discovery in Databases process. Adapted from Fayyad et. al [3].	1
1.2	Types of data irregularities. Adapted from Das et. al [5].	2
1.3	Examples of distribution-based irregularities [5]. The ideal decision boundary represents the optimal limit between the two classes and the learned decision boundary corresponds to the one more likely to be learnt by a linear classifier.	4
2.1	Methods to handle the missing data. Adapted from García-Laencina et. al [12].	9
2.2	Examples of overlapping classes and noise.	12
2.3	Illustration of how to generate artificial data using Synthetic Minority Oversampling Technique (SMOTE). Adapted from Fernández et. al [27].	14
2.4	Example of the small disjuncts problem within the minority class. . .	15
2.5	Example of the four types of samples of the minority class with $k = 5$. . .	22
3.1	flower	32
3.2	flower-min-imbalanced	32
3.3	two-circle-integumental	32
3.4	two-circle-integumental-min-imbalanced	32
3.5	paw3	32
3.6	subclus5	32
3.7	Distribution of each type of sample in real world datasets.	33
3.8	Types of data irregularities. Adapted from Das et al. [5]	35
3.9	Imputation error of the artificial datasets using k-Nearest Neighbours (k-NN) with 20% of missing values.	36
3.10	Imputation error of the artificial datasets using Multiple Imputation by Chained Equations (MICE) (with Linear Regression as estimator) with 20% of missing values.	36

3.11	Imputation error using k-NN of datasets with different Imbalance Ratio (IR) and 20% of missing values.	37
3.12	Imputation error using MICE (with Linear Regression as estimator) of datasets with different IR and 20% of missing values.	37
3.13	Pearson Correlation Coefficient between the imputation error and some datasets' characteristics.	38
3.14	Pearson Correlation Coefficient between the imputation error and the complexity metrics.	39
3.15	Types of data irregularities. Adapted from Das et al. [5]	40
3.16	Pearson Correlation Coefficient between the imputation error and the distribution of the minority class.	41
3.17	Pearson Correlation Coefficient between the F1-score and the datasets characteristics.	46
3.18	Pearson Correlation Coefficient between the F1-score and the complexity metrics.	47
3.19	Pearson Correlation Coefficient between the F1-score and the distribution of the minority class.	47
3.20	Correlation between <i>diff</i> and some datasets characteristics.	49
3.21	Sum of squared distances from each point to its assigned center for $k \in [2, 10]$	50
C.1	Imputation error of the artificial datasets using k-NN with 5% of missing values.	69
C.2	Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 5% of missing values.	69
C.3	Imputation error of the artificial datasets using k-NN with 10% of missing values.	69
C.4	Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 10% of missing values.	70
C.5	Imputation error of the artificial datasets using k-NN with 40% of missing values.	70
C.6	Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 40% of missing values.	70
D.1	Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 5% missing.	72
D.2	Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 10% missing.	73
D.3	Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 40% missing.	74

E.1	Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 5% missing.	76
E.2	Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 10% missing.	77
E.3	Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 40% missing.	78

This page is intentionally left blank.

List of Tables

1.1	Classifiers assumptions about the data and problems that arise when they are violated. Examples of Class Imbalance, Small Disjuncts and Class distribution skew are represented in Figure 1.3.	3
2.1	Example of a conventional approach to interpret the absolute value of the Correlation Coefficient ($ \rho $). Adapted from Schober et al. [35].	17
2.2	Other example of a conventional approach to interpret the absolute value of the Correlation Coefficient ($ \rho $) presented by Taylor et al. [34].	17
2.3	Confusion matrix. Adapted from Monard et al. [36].	17
2.4	Summary of the literature review	28
3.1	Real datasets characteristics	31
3.2	F1-score results of baseline, Multiple Imputation-Based Minority Over-sampling Technique (MI-MOTE) and modified MI-MOTE. The best and second best results are in bold and underlined, respectively. . . .	45
3.3	Mean values of the datasets characteristics for each cluster.	52
3.4	Mean values of the complexity metrics for each cluster.	52
3.5	Mean values of the percentage of each type of point for each cluster. .	52
3.6	Mean imputation error and F1-score for each approach and cluster. .	53
A.1	Complexity metrics description. Adapted from <i>pymfe</i> [53].	65
B.1	Properties of the datasets used in the experiments.	67

This page is intentionally left blank.

Chapter 1

Introduction

Over the years, the evolution of the technology increased exponentially the amount and the complexity of the available data. Also, the growing use of Machine Learning (ML), specially deep learning techniques demand more data to optimize their results [1].

In the late 1980s, Piatetsky-Shapiro named a workshop held at the *International Joint Conference on Artificial Intelligence* as Knowledge Discovery in Databases (KDD) [2]. In 1990, Fayyad et al. [3] defined KDD as the “overall process of discovering useful knowledge from data” [3]. This process consists of five steps (Figure 1.1): Selection, Preprocessing, Transformation, Data Mining and Interpretation/Evaluation.

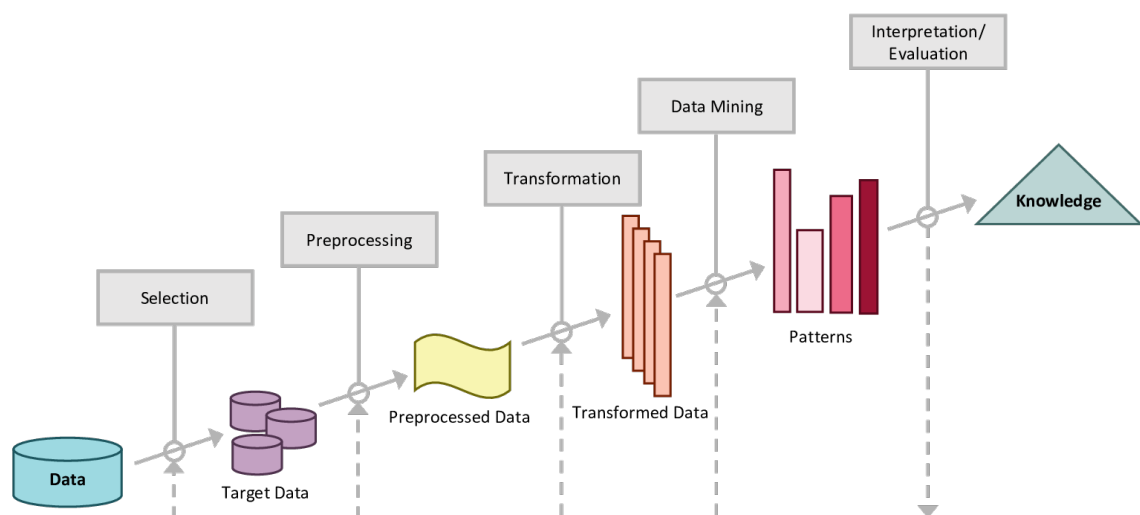


Figure 1.1: Knowledge Discovery in Databases process. Adapted from Fayyad et. al [3].

The first step includes understanding the problem at hand and its goal, collecting

relevant prior knowledge and the necessary data. The next step is cleaning and preprocessing the data, *i.e.*, remove noise and outliers, deal with missing values, among others. In the third step, it is performed dimensionality reduction or other useful transformation techniques to represent the data. In the fourth step, a data mining approach is selected to reach the goals defined in the first step. The last step consists in analysing and validating the results obtained in the previous phase.

Many factors can affect the performance of a ML model based approach and the quality of the data is first and foremost [4]. The increasing complexity brought some data quality problems that lead to poor classification performance. Therefore, the preprocessing phase will occupy an important role in the performance of such models.

1.1 Context and Motivation

Data irregularities have a high impact on the performance of ML algorithms. Data irregularities are essentially situations where the distribution of the data or the features deviate from what could have been ideal, being biased, skewed or incomplete [5].

Most of the traditional classifiers make a few assumptions about the data [5]. When these assumptions are violated, some problems arise. These assumptions and respective problems are described in Table 1.1.

The problems aforementioned can be divided in two categories: distribution-based and feature-based, as shown in Figure 1.2.

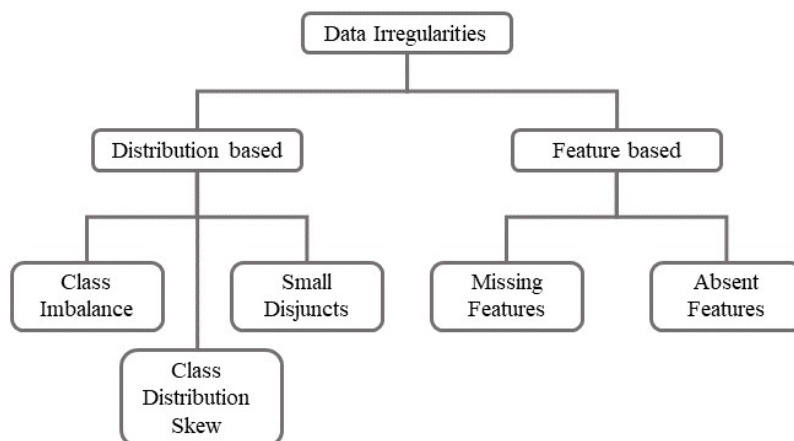


Figure 1.2: Types of data irregularities. Adapted from Das et. al [5].

A large amount of research works have already studied these five problems separately

Assumption	Problem	Problem Description
Data is equally distributed	Class Imbalancement	One or more classes are underrepresented
Each class is equally distributed	Small Disjuncts	Sub-concepts within classes are underrepresented
All the classes have similar class-conditional distributions	Class distribution skew	Different classes possess very different class-conditional distributions
Feature values are all defined	Absent Features	Some features are undefined for some of the data points due to its nature
Feature values are all known	Missing features	Corruption of feature values due to noise, equipment malfunction, etc

Table 1.1: Classifiers assumptions about the data and problems that arise when they are violated. Examples of Class Imbalance, Small Disjuncts and Class distribution skew are represented in Figure 1.3.

but the interrelations between them are an important matter to discuss, since more than one assumption can be violated at the same time. In [6, 7], the authors studied the connection between the class imbalance and small disjuncts problems. The authors of the first article concluded that the prediction of small disjuncts of the majority class is more accurate than the ones of the minority class with the same size. In the second paper, the authors concluded that the small disjuncts problem has a greater impact than the class imbalance on the decrease in accuracy.

Das et al. [5] concluded that there are none research works about the connection between distribution-based and feature-based data irregularities. Missing values and absent features are independent and intrinsic to the dataset. The interrelations between distribution-based and feature-based irregularities have not been studied yet. The surveys on this topic only propose new approaches to deal with these problems individually, neglecting the effect irregularities from one category have on the other. For example, class imbalance and missing data are two problems that, most of the times, happen together in real datasets. In his work, Das et al. [5] left some open issues about the connection between these two types of data irregularities:

- What is the effect of missingness on the performance of classifiers designed to handle distribution-based irregularities and vice-versa?
- When does missingness arise distribution-based irregularities?

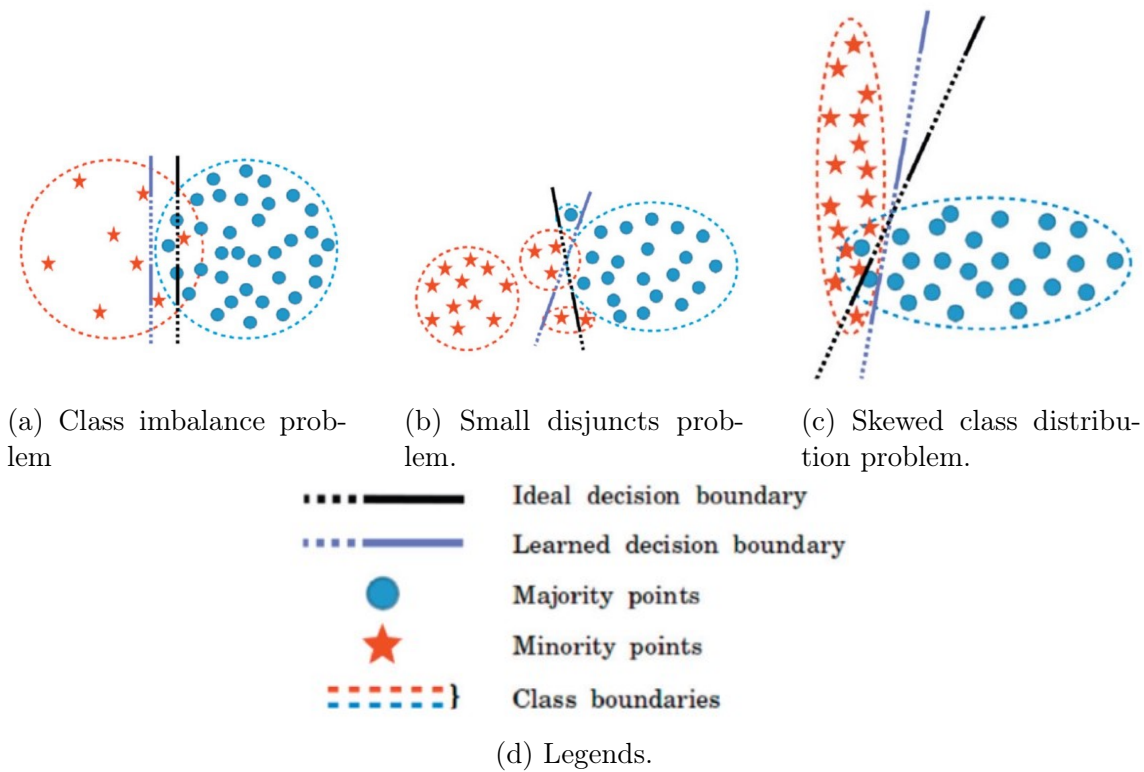


Figure 1.3: Examples of distribution-based irregularities [5]. The ideal decision boundary represents the optimal limit between the two classes and the learned decision boundary corresponds to the one more likely to be learnt by a linear classifier.

1.2 Main Goals

Addressing the issue highlighted in the previous section, the main goal of this thesis is **analysing the interrelation between missing data and class imbalance and their impact on the classification performance**. To research it, some hypothesis were formulated:

- What effect does the increasing of missingness has on the classification on imbalanced scenarios?
- What impact the class imbalance has on the missing imputation and on the performance of the classification?

1.3 Document Structure

This thesis is structured as followed: in Chapter 2, important concepts to understand the performed work are described and a literature review on the topics of class imbalance and missing data is provided; in Chapter 3 it is analysed the experiments

performed in this thesis and the results obtained; the main conclusions and future work are presented in Chapter 4.

This page is intentionally left blank.

Chapter 2

State of the Art

With the increasing importance and complexity of data pipelines, data quality became one of the key challenges in modern software applications [8]. Data quality has different definitions and interpretations. This concept is mainly researched in two fields: databases, where it is studied from a technical point of view, and management, where other aspects about the data are concerned, such as dimensions, accessibility, relevancy, interpretability, etc [9]. This thesis is focused on databases data quality problems.

Poor data quality decreases significantly the performance of the ML algorithms. Two of the most frequent real-world data quality problems are missing data (absence of information) and class imbalance (at least one of the classes is underrepresented). Statistical and machine learning models generally need complete data [10] and if a dataset is imbalanced the prediction or classification model can be biased towards the majority class [11].

In this chapter, the fundamental notions needed to understand the analysis performed in this work are explained. In the last section, some literature about the topic in hand is reviewed.

2.1 Missing Data

Missing data is the absence of information in a dataset and is a common problem in real-world datasets. For example, in the UCI repository (which is one of the most frequently open source used repositories) more than 45% of the datasets present missing data [12]. It can occur due to erroneous inputting of data, incorrect measurements, malfunctioning measuring equipment, non-response in surveys, among

other reasons. For example, when a student is filling a psychology questionnaire, he might not answer one question because he didn't see it or because it doesn't make sense to him (for example, if the question is "How many kids do you have?" and he doesn't have one, the student pass this question).

2.1.1 Missing Data Mechanisms

The type of mechanism by which the data is missing affects certain assumptions made when solving this problem. There are three types of missing data: **Missing Completely At Random (MCAR)**, **Missing At Random (MAR)** and **Missing Not At Random (MNAR)**. To explain them, consider the dataset $X = \{X_{obs}, X_{miss}\}$, where X_{obs} and X_{miss} are the observed and missing parts of the data, respectively, and the missing matrix R where the location of the missing values is indicated. The missing mechanism can be represented by the probability of a sample being missing $P(R)$ given the observed and missing samples defined in Equation 2.1.

$$P(R|x_{obs}, x_{miss}) \tag{2.1}$$

The three mechanisms will be explained using the process of responding to a survey as an example.

Missing data is considered **MCAR** when the probability of a value being missing does not depend on the value itself nor the observed values on the other variables. The probability of a value being missing only depends on itself, therefore, $P(R|x_{obs}, x_{miss}) = P(R)$. Using the example of the survey, when someone skips a question because he didn't see it, the value is MCAR because it does not depend on the answers to the other questions of the survey.

MAR occurs when the cause of the missing data is related to other variables of the dataset but not with the values that would be in that missing data. The probability that some data is missing can be defined as $P(R|x_{obs}, x_{miss}) = P(R|x_{obs})$ Considering one of the questions of the survey is "How much do you weight?", a woman might not answer this question, not because of the actual answer, but because a person of the female gender usually don't feel comfortable to give this information. The weight is missing depending on the variable "gender" but not because of the value itself.

When the reason for missing data depends on missing and observed values, the mechanism is **MNAR**. This mechanism cannot be determined since it depends on

unobserved data. Considering the question “How many cigarettes do you smoke per day?”, someone that answered “Yes” to the question “Do you smoke?” and smokes a lot might not answer the first question because he wants to hide it. The missingness of the data depends on another variable and on the value missing.

2.1.2 Missing Data Imputation

There are several ways to handle missing data. Figure 2.1 resumes the different type of approaches to deal with missing data. Imputation is the most commonly used [12], therefore, this work will focus on this type of approaches.

Imputation methods try to replace a missing value by plausible ones and are mainly divided into statistical-based or machine learning based [12, 13]. Statistical methods replace the missing values with the most similar ones without building a model to find their similarity (e.g. Mean/Median imputation, Multiple Imputation by Chained Equations (MICE)). Machine learning-based methods construct a predictive model to estimate the missing values (e.g. k-Nearest Neighbours (k-NN) imputation and Support Vector Machines (SVM) imputation).

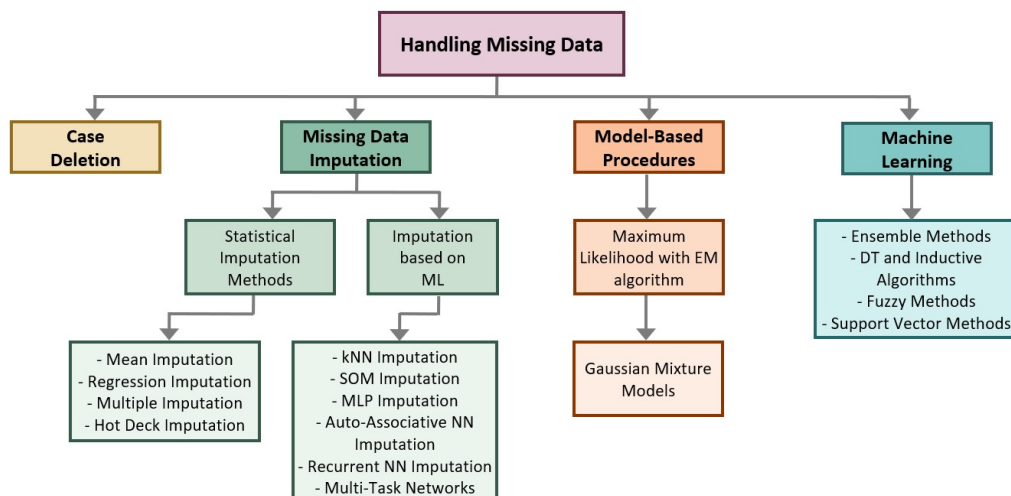


Figure 2.1: Methods to handle the missing data. Adapted from García-Laencina et. al [12].

Mean/Median imputation

Mean/Median imputation are the simplest statistical-based imputation methods and they replace the missing values by the mean or median of the variable, respectively [10]. Mean imputation is more robust in the presence of outliers. The disadvantages

of using this approaches are that they doesn't consider the correlation between variables and can produce biased estimates.

Multiple Imputation by Chained Equations

MICE is a particular multiple imputation technique that can be explained in four steps [14]:

1. Replace the missing values with a simple imputation method, such as mean or median;
2. The imputations in Step 1 are set back to missing for one variable (*var*);
3. The observed values from *var* are regressed on the other variables, *i.e.*, the other variables are the independent variables in the regression models and *var* is the dependent variable;
4. The missing values for *var* are replaced with predicted ones from the regression.

This iterative process through the missing features is repeated until the convergence of the imputation parameters (e.g., coefficients of the regression model). In Step 3, the regression model can be a linear regression, Random Forest (RF) regression, logistic regression, among others. At the end of the process, the missing values have been replaced by values that reflect the relationships between the data.

k-Nearest Neighbours

k-NN is a popular classification method that can be used for imputation of missing values. Given an incomplete instance, this method selects the k nearest complete instances and estimates the missing values with the mean or weighted mean (for continuous features) or mode (for categorical features) [15]. The weighted mean attributes weights to the neighbours regarding their distance to the incomplete sample.

This method requires the selection of the optimal number of neighbours k and the distance metric between the incomplete samples and their neighbours [16]. The distance metric should be chosen taking into account the variables nature (categorical or numerical). The Euclidean distance is one of the most popular distance metric for numerical features and is defined in Equation 2.2, where x are samples from the datasets and n is the number of features.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.2)$$

Random Forest

RF imputation is an iterative imputation method [17]. This approach has some advantages when compared with other popular imputation methods. RF imputation [18]:

- Is capable of handling different types of features;
- Addresses the nonlinearity of features;
- Scales to high dimensions while avoiding overfitting;
- Does feature selection.

Recently, an approach called missForest was proposed by Stekhoven et al. [17]. In this method, the missing data problem is considered a prediction problem. Values are imputed by regressing one feature at a time against the other features. Then, the missing data in the dependent variable (feature being regressed) are imputed using the fitted forest.

2.2 Distribution-based irregularities

Distribution-based irregularities englobe three main data problems: class imbalance, small disjuncts and class distribution skew. This thesis is mainly focused on the first two problems, therefore, they will be further explained in this section.

2.2.1 Class Imbalance

Imbalanced data occurs when there's a significant imbalanced distribution between classes of a dataset [19, 20, 21]. In a binary scenario, the dataset is imbalanced if one of the classes (minority class) is underrepresented compared to the other class (majority class). This problem can significantly compromise the performance of most standard learning algorithms [19, 22]. Some fields where imbalanced data is present are, for example, medical diagnosis prediction of rare diseases, fraud detection in transactions, detection of network intrusions, among others [21].

There are some metrics used to measure how imbalanced a dataset is. The Imbalance Ratio (IR) is widely used to measure it. Considering n_{maj} and n_{min} the number of samples in the majority and minority class, respectively, the IR is defined on Equation 2.3. This value represents the number of majority samples in the dataset per each minority sample, *i.e.*, if $IR = 10$, for each minority sample exists 10 samples from the majority class. A dataset is considered imbalanced if $IR > 2$ [23].

$$IR = \frac{n_{maj}}{n_{min}} \quad (2.3)$$

Class imbalance has proven to be a challenging problem, but can be severally worsened when combined with other data difficulty factors, such as [24]:

- **Small Disjuncts:** Small meaningful clusters of the minority class far from the class's centroid, *i.e.*, the minority class is represented in smaller clusters (Figure 1.3b);
- **Overlap:** Majority and Minority samples are in the same feature space, *i.e.*, have the same have very similar feature values while belonging to different classes (Figure 2.2);
- **Noisy Data:** Presence of non-meaningful instances that degrade the performance of the learning algorithms (Figure 2.2).

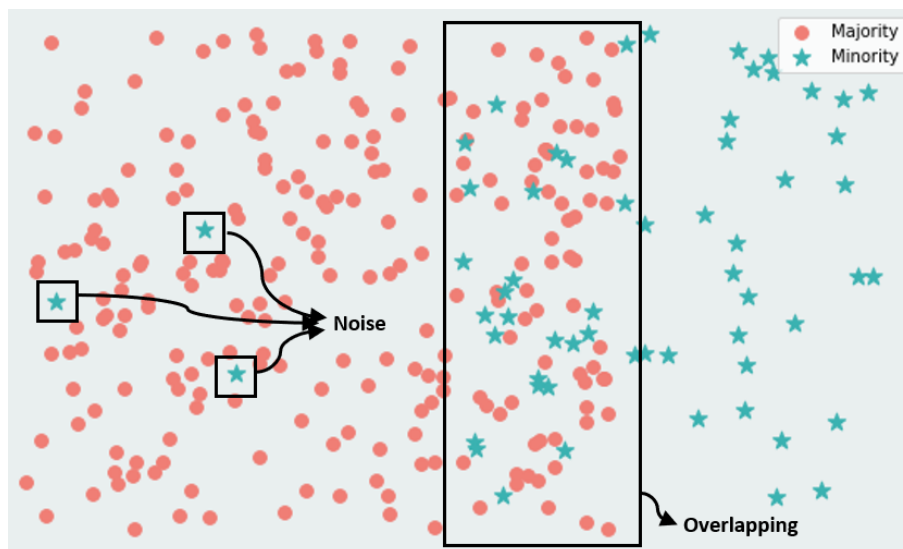


Figure 2.2: Examples of overlapping classes and noise.

The imbalance can be handled following different approaches that can be categorized into two groups [20]:

- Data-level, where the data is preprocessed and the data distribution is altered;
- Algorithmic-level, where the classifiers are modified to handle the imbalance.

Data-level approaches are the most common, as they have proven to be efficient, simple to implement and do not depend on the classifier [20]. This type of strategies use two sampling techniques: undersampling, that removes samples from the majority class, and oversampling, where the minority class is replicated.

Oversampling techniques generate synthetic data to increase the size of the minority class. Synthetic Minority Oversampling Technique (SMOTE) is a commonly used benchmark for oversampling technique [25].

Synthetic Minority Oversampling Technique

SMOTE creates artificial data taking into account the space similarities between minority samples [19]. The minority class is oversampled by taking the k nearest minority class neighbours of each minority sample x_i and creating synthetic examples between x_i and some of the random neighbours [26]. There are some extensions of this algorithm [27], like Borderline-SMOTE (only the borderline samples are considered for oversampling), Safe-Level-SMOTE (the opposite to Borderline-SMOTE, *i.e.*, only examples around safe regions are synthesized), among others.

For example, if each minority sample has to be oversampled four times, SMOTE selects 4 random samples from the k nearest neighbours and generates a sample x_{new} following Equation 2.4, where $\delta \in [0, 1]$ is a random number, x_i is the sample to be oversampled and $x_k, k \in [0, 1, 2, 3]$ are the chosen neighbours. Figure 2.3 illustrates how synthetic data is generated using this approach.

$$x_{new} = x_i + \delta(x_k - x_i) \quad (2.4)$$

Over time, researchers observed that SMOTE produces noise by choosing samples randomly [25]. Therefore, they tried to create other oversampling techniques based on SMOTE to decrease the generation of wrong artificial samples.

Batista et al. [28] proposed a method called SMOTE+ENN where first they oversample the minority class using SMOTE and then use Wilson's Edited Nearest Neighbor Rule (ENN) to remove samples from both classes that differ from the neighbourhood. Any example whose class differs from at least two of its three closest neighbours is removed from the training set. This approach provided good results, specially for datasets with a high IR [25].

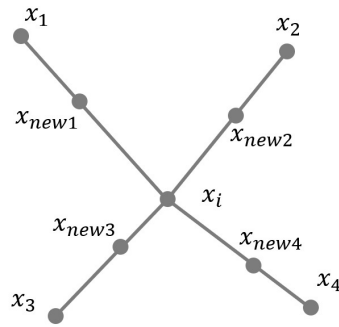


Figure 2.3: Illustration of how to generate artificial data using SMOTE. Adapted from Fernández et. al [27].

2.2.2 Small Disjuncts

A disjunct is a conjunctive definition of a subconcept of the original concept [29]. The size of each disjunct corresponds to the number of samples that are correctly classified. Therefore, a small disjunct is simply a disjunct with a small coverage, which is represented in Figure 2.4. In this figure, the minority class is divided in two small disjuncts.

Small disjuncts often have rare examples, small number of training examples in the feature space [30]. The difference between rare and noisy samples is that the first type is a valid concept while the later do not have a physical meaning. Rare examples form small disjuncts that are underrepresented subconcepts of the minority class [29].

Since small disjuncts are less represented, their classification error will be higher than the larger disjuncts. The set of assumptions made by the classifiers will not take into account the disproportional class distributions.

When considering a binary problem, the minority class is more likely to create small disjuncts since it has less examples [31]. When a classifier is generalizing, common cases might impose over rare examples, favoring larger disjuncts.

Some authors have already studied some relationships between the small disjuncts problem and class imbalance. Jo et al. [32] showed that high imbalanced datasets might have a higher number of small disjuncts. Quilan [33] proved that the small disjuncts in the minority class generates a higher error than the ones in the majority class.

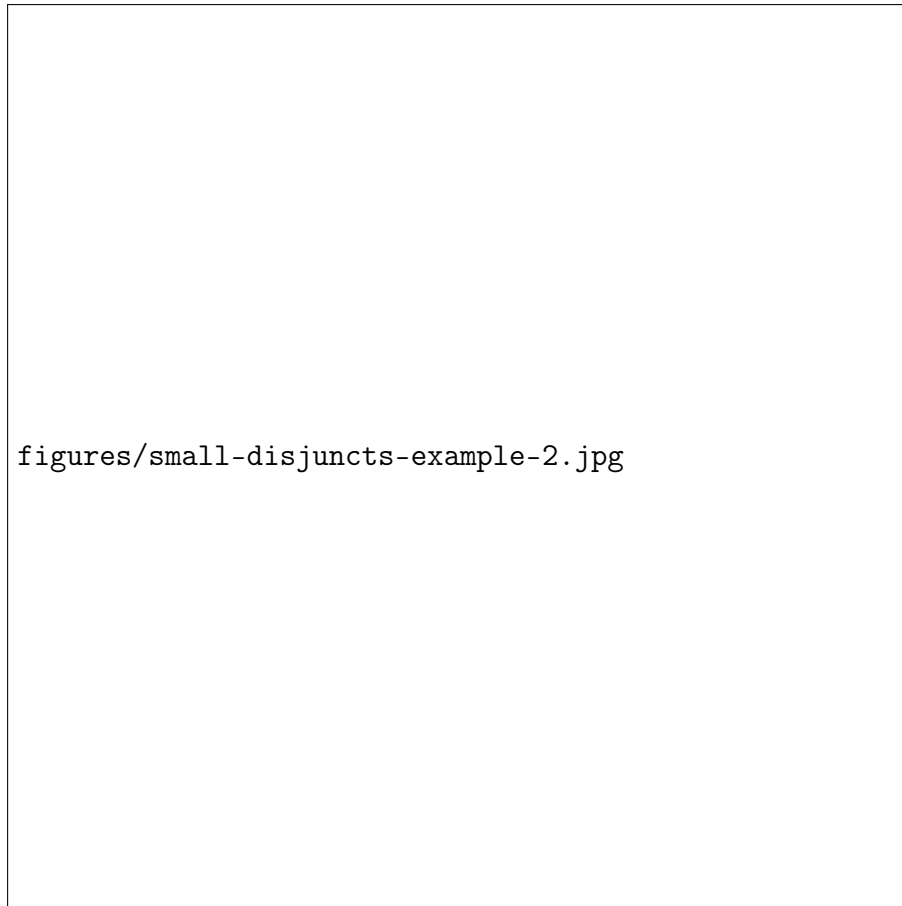


Figure 2.4: Example of the small disjuncts problem within the minority class.

2.3 Performance Evaluation Metrics

In this section, several metrics to evaluate the performance of the algorithms are described, focusing on metrics that are more suitable to evaluate missing imputation and classification on imbalanced scenarios.

2.3.1 Imputation Quality

The performance of an imputation method can be measured through the Classification Error (CE), *i.e.*, the best method is the one that minimizes the classification error, or comparing the original values and the imputed ones [12]. In the first approach, the imputation that minimizes the CE can affect the data distribution, specially if the same method is used in data with different distributions. Therefore, the second method is more adequate to evaluate the imputation performance. Let's consider x the original values of a certain feature, \hat{x} the imputed values, \bar{x}_i the mean of the original values, $\bar{\hat{x}}$ the mean of the imputed values and n the number of missing

values in that feature.

Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) (Equation 2.5) is a quadratic metric used to measure the difference between two features. Although utilized in several studies in several studies to compare imputation methods, this metric might not be appropriate for large differences between the original and the imputed values.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (2.5)$$

Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) (Equation 2.6) is less affected by large errors and is widely used to measure the imputation performance.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x - \hat{x}| \quad (2.6)$$

Pearson Correlation Coefficient (ρ) and Coefficient of Determination (R^2)

The Coefficient of Determination (R^2) is equivalent to the square of Pearson Correlation Coefficient (ρ). This metric measures the correlation between two features that, when considering the performance of the imputation, are the feature with the original values and the feature with the imputed values. ρ and R^2 are defined in Equations 2.7 and 2.8, respectively, where X and Y are the variables to be compared, μ_X and μ_Y are the mean of X and Y , respectively, σ_X and σ_Y are the standard deviation of X and Y , respectively, and E is the expectation (mean). These two measures are between 0 and 1.

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (2.7)$$

$$R^2 = \rho^2 \quad (2.8)$$

Tables 2.1 and 2.2 present a possible interpretation for the Correlation Coefficient

values [34, 35].

Intervals of $ \rho $	Interpretation
$[0.0, 0.1[$	Negligible correlation
$[0.1, 0.4[$	Weak correlation
$[0.4, 0.7[$	Moderate correlation
$[0.7, 0.9[$	Strong correlation
$[0.9, 1.0]$	Very strong correlation

Table 2.1: Example of a conventional approach to interpret the absolute value of the Correlation Coefficient ($|\rho|$). Adapted from Schober et al. [35].

Intervals of $ \rho $	Interpretation
$[0.0, 0.36[$	Weak correlation
$[0.36, 0.68[$	Moderate correlation
$[0.68, 0.9[$	Strong correlation
$[0.9, 1.0]$	Very strong correlation

Table 2.2: Other example of a conventional approach to interpret the absolute value of the Correlation Coefficient ($|\rho|$) presented by Taylor et al. [34].

2.3.2 Imbalance Classification Quality

In a binary classification, consider the following notation for a classification output, where the positive class is the minority class: True Positives (TP) (samples correctly predicted as positive), False Positives (FP) (samples predicted as positive, but in reality are negative), True Negatives (TN) (samples correctly predicted as negative) and False Negatives (FN) (samples predicted as negative, but in reality are positive). Table 2.3 represents a confusion matrix. This matrix summarises the type of errors described before.

	Positive Prediction	Negative Prediction
Positive Class	TP	FN
Negative Class	FP	TP

Table 2.3: Confusion matrix. Adapted from Monard et al. [36].

In the following sections, some popular performance measures will be described. The first one, the accuracy, is not suitable for imbalanced scenarios whereas precision, recall, F1-score and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) are popular metrics for classification problems with imbalanced data.

Accuracy

The accuracy (Equation 2.9) of a model is one of the most popular metrics to evaluate the performance of the classifier, but it is not suitable when the data is imbalanced [37]. For example, consider a dataset with 1000 samples, where 980 are from the negative class and 20 from the positive one ($IR = 49$). A simple classifier can classify all the examples as negative and have an accuracy of 98%, when, in reality, this method cannot predict correctly the outcome of samples from the minority class. Therefore, the accuracy is not a suitable metric for imbalanced situations.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.9)$$

Precision, Recall and F1-score

The precision, also known as Positive Predictive Value (PPV), is the proportion of samples correctly classified as positive among all samples that are predicted as positive (Equation 2.10).

$$precision = \frac{TP}{TP + FP} \quad (2.10)$$

On the other hand, the recall, also referred to as Sensitivity or True Positive Rate (TPR), is the proportion of positive samples correctly predicted (Equation 2.11). This metric is an indicator of the performance of classifying correctly the minority class.

$$recall = \frac{TP}{TP + FN} \quad (2.11)$$

In medical classification, the main goal is to improve the recall (not miss patients with diseases) without compromising the precision (diagnose all the patients with a disease). F-score is a metric that unifies the previous two and is represented in Equation 2.12, where $\beta \in [0, 1]$ is a weight that translates the importance assigned to the recall.

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall} \quad (2.12)$$

When $\beta = 1$, *i.e.*, the recall is as important as the precision, this metric is called

F1-score and Equation 2.12 is simplified to Equation 2.13.

$$F_1 = 2 \times \frac{\textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \quad (2.13)$$

G-Mean

Geometric Mean (G-Mean) is another metric that is independent of the classes distribution [38] and is defined in Equation 2.15, where the Specificity is the proportion of negative instances correctly predicted (Equation 2.14) and the Sensitivity is the recall. This measure tries to maximize the accuracy on each class while keeping these accuracies balanced.

$$\textit{specificity} = \frac{TN}{TN + FP} \quad (2.14)$$

$$G\text{-Mean} = \sqrt{\textit{sensitivity} \times \textit{specificity}} \quad (2.15)$$

ROC and AUC

AUC makes use of the ROC curve to exhibit the trade-off between the TPR and the False Positive Rate (FPR), *i.e.*, $(1 - \textit{specificity})$ versus *sensitivity*. It is expected that when the TPR increases, the FPR increases therefore the *specificity* decreases.

A satisfactory ROC curve is supposed to lie above the identity line, that represents the scenario of randomly guessing the class. The ideal point on the ROC curve is at the top left, where the TPR is 1 and the FPR is 0, in other words, the classifier predicts correctly all the positive samples and no negative sample is misclassified as positive [37]. In this case $AUC = 1$.

The AUC measures the ability of a classifier to separate the positive class from the negative one and is used as a summary of the ROC curve. This scalar is the integration of this curve and $AUC \in [0, 1]$. If $0.5 < AUC < 1$, there is a high chance the classifier will be able to distinguish the positive samples from the negative ones.

2.4 Datasets characteristics

The analysis of the datasets characteristics allows to understand in which scenarios a ML algorithm might fail [39]. This analysis can focus on many aspects about

a dataset, but this thesis will only focus on the complexity of a dataset and the distribution of the minority class.

2.4.1 Meta-features

Lorena et al. [39] defined some metrics that measure how complex a dataset is. The authors divided all measures into six categories: feature-based, linearity, neighborhood, network, dimensionality and class imbalance.

In this section, the metrics used in this thesis will be described. These metrics were chosen after analysing which ones made more sense for the problem and described better the data. Some metrics were contradictory and were also not used in this work. All measures are between 0 and 1 and the higher the values, the more complex the data is. Table A.1 has a description of each individual metric.

Feature-based Measures

Feature-based measures evaluate the discriminating power of the features. If the dataset has at least one highly discriminating feature, the problem is simpler. In these work, five feature-based metrics were used: Maximum Fisher's Discriminant Ratio ($f1$), Directional-vector Maximum Fisher's Discriminant Ratio ($f1v$), Volume of overlapping ($f2$), Maximum Individual Efficiency ($f3$) and Collective Feature Efficiency ($f4$). These measures mainly study the amount of overlap between different classes.

Measures of Linearity

This type of measures study to what extent the classes are linearly separable, *i.e.*, if it is possible to separate them with an hyperplane. A more linearly separable problem is considered simpler. Three linearity measures were analysed in this thesis: Sum of the Error Distance by Linear Programming ($l1$), Error of Linear Classifier ($l2$) and Non-Linearity of a Linear Classifier ($l3$). These metrics measure the error of linear and non-linear classifiers.

Neighbourhood Measures

These type of measures study the decision boundaries and characterize the class overlap by analysing the local neighbourhood of each sample. Some measures also

characterize the internal structure of each class. All metrics use a distance metric calculated with a heterogeneous distance measure. The four metrics used in these work are: Local Set Average Cardinality (*lsc*), Fraction of Borderline Points (*n1*), Error Rate of the Nearest Neighbor Classifier (*n3*) and Non-Linearity of the Nearest Neighbor Classifier (*n4*).

Network Measures

In these measures, the dataset is represented as a graph. These graph must maintain the similarities and distances between samples to ensure the data relationships. Two nodes are connected if the distance between them is lower than a certain threshold. The two metrics used are: Average Density of the Network (*density*) and Clustering Coefficient (*cls_coef*).

2.4.2 Minority class distribution

Some authors have already shown that the class imbalance alone might be harmless for the classification algorithms but, when combined with other data quality problems, such as small disjuncts, it might have a negative effect on the recognition of the minority class [40]. Napierala et al. [41] proposed a way to distinguish different types of minority samples. The authors defined four types of examples:

- *Safe*: Examples situated in the homogeneous regions populated by one class only. These type of points are easier to classify;
- *Borderline*: Samples that are located in the regions around decision boundaries between classes;
- *Outlier*: Rare but valid sub-concept that should not be mistaken as a noisy sample;
- *Rare*: Pairs or triples of minority samples located in the majority class region. They are far from the boundary region but also are not isolated, such as the outliers.

Each sample is labelled following a neighbourhood-based approach because the minority class often has smaller concepts, therefore, a local analysis is better suited than a global one. The number of neighbours from the same class is what defines each type of sample. Consider the notation $n_{kmin} : n_{kmaj}$, being n_{kmin} and n_{kmaj} the

number of minority and majority samples in the neighbourhood, respectively, and k the size of the neighbourhood. Figure 2.5 represents a dataset with the four types of minority samples defined before with $k=5$. In this example, if the proportion of minority and majority samples are:

- 5 : 0 or 4 : 1, the sample is labelled **safe**;
- 3 : 2 or 2 : 3, the sample is labelled as **borderline**;
- 1 : 4, the sample is labelled as **rare**;
- 0 : 5, the sample is labelled as **outlier**.

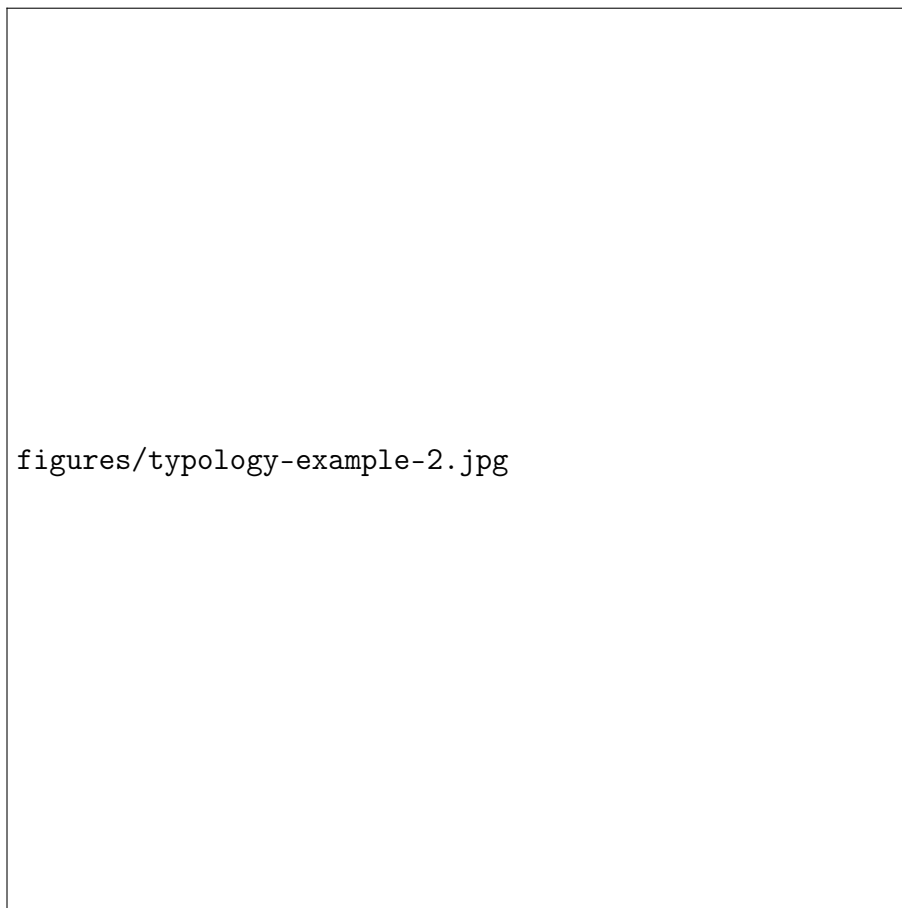


Figure 2.5: Example of the four types of samples of the minority class with $k = 5$.

Saéz et al. [42] used this characterization to study if preprocessing only certain types of minority samples improves the classification performance. They concluded that this approach had a significantly improvement over standard preprocessing methods. Additionally, they showed that in most datasets, oversampling the borderline examples improved the results and the rare samples should be oversampled if the percentage of safe examples is low.

2.5 Literature Review

In this section, approaches used to solve the problems of missing data and class imbalance when they occur simultaneously are reviewed. This problem can be solved following two different approaches: sequentially, described in Section 2.5.1, or simultaneously, overviewed in Section 2.5.2.

2.5.1 Sequential Approaches

The conventional approach to address missing data and class imbalance when they occur simultaneously is solve each one of the problems individually. The missing mechanism considered by the authors in this section was MCAR.

The main goal of Wang et al. [16] was to develop an algorithm with high precision to classify Diabetes Mellitus on an imbalanced and incomplete dataset. The chosen dataset was the Pima Indian Diabetes Database (PIDD), where only 392 of the 768 are completed and the $IR = 1,87$. They first compensated the missing values with a Naive Bayes (NB) approach and then oversampled the minority class with an Adaptive Synthetic Sampling Method (ADASYN). For the classifier, the authors decided to use a RF. To evaluate their approach, they compared it to other benchmark machine learning algorithms as the classifiers (NB, SVM and Decision Tree (DT)) using different metrics (accuracy, precision, recall, F1-score and AUC). The authors also compared their approach to a selection of existing ones proposed by other authors considering only the accuracy. The authors concluded that the proposed method outperforms the compared algorithms, getting an accuracy of 87,10%, 2,39% higher than the second best approach.

Liu et al. [43] proposed a method to help prevent cerebral strokes. To eliminate dimension relationships, the data is normalized with Z-score. The authors used a RF Regression (RFR) to perform the imputation if there is a high correlation between features. Otherwise, the missing values are imputed using statistical methods, like mean. To predict if a patient will have a cerebral stroke, a Deep Neural Network (DNN) based on Automated Hyperparameter Optimization (AutoHPO) model is used. In this algorithm, the majority class is undersampled. A Principal Component Analysis (PCA) is applied to remove irrelevant features and a K-means is used to perform a preliminary classification for AutoHPO. The elbow method is used to find the optimal k . The dataset has an $IR \approx 55.3$ and missing rates of 0.3 and 0.03 for the smoking status and BMI, respectively. Accuracy, specificity, sensitivity, G-Mean, False Negative Rate (FNR) and FPR are the metrics used to

compare the proposed method with baseline ones. Since the smoking status as low correlation with the other features and commonly used prediction methods have low accuracy when predicting this variable, a statistical imputation is adopted for this feature. BMI has a higher correlation with the other features, therefore, RFR is used for the imputation and its hyperparameters are optimized through grid search. The proposed method is compared with a DNN, Bagging, RF, Adaptive Boosting (AdaBoost) and XGBoost. The method described in this paper obtained the lowest FPR and higher G-Mean. The FPR is slightly higher, but it is in an acceptable range. The authors concluded that their approach can improve the FNR without a large cost of the accuracy (1.7% higher than the mean of the compared methods). Future work can focus on feature analysis based on DNN, such as sensitivity analysis and l_1 regularization.

Ozan et al. [44] proposed a solution for the machine learning problem challenge in IDA 2016, where they had to solve a binary classification problem on an incomplete and imbalanced dataset. The authors proposed a k-NN based approach, where first they estimated the missing values with the mean of the k closest samples that have the attribute complete, and then, they classified all the samples with a weighted k-NN, where they calculated the probability that a sample belongs to each class. The neighbours weights are attributed linearly between $[0, 1]$. In order to compensate the class imbalance, rather than perform undersampling or oversampling, examples from the minority class are given a higher weight. The best parameters are found using the stochastic gradient descent approach. The k-NN distance metric used was a slightly modified version of the Heterogeneous Euclidean Overlap Metric (HEOM). The dataset has numerical attributes and histograms, the $IR = 59$ and the missing rate is not specified. The performance of this method was compared with other baseline approaches, such as SVM, RF and AdaBoost, using a cost function defined by IDA 2016 that penalizes more a missed positive (FN). The authors performed 5 different tests with 5-fold cross-validation and concluded that the proposed method outperformed the baselines ones.

Puri et al. [45] analysed 84 different models in noisy incomplete and imbalanced datasets. Their study was divided in three parts: imputation of incomplete and imbalanced datasets, resampling techniques for noisy and imbalanced datasets, combination of the previous two. The authors considered 29 binary completed datasets with IR between $[1.82, 58.40]$. Feature noise is first introduced by corrupting the data by randomly assigning a value between the minimum and maximum of a particular feature. The noise level tested was 0%, 10% and 20%. Then, missing values are generated MCAR and the missing rate considered was 0%, 5%, 10%, 15% and 20%. The missing data imputation techniques chosen were Expectation-

Maximization (EM), MICE, k-NN, mean and median. To deal with class imbalance, 13 types of SMOTE were used: SMOTE, SMOTE-PSO, SMOTE-IPF, SOI-CJ, SMOTE-ENN, DBSMOTE, SMOTE-TomekLink, SMOTE-OUT, GASMOTE, NRAS-SMOTE, AND-SMOTE, NRSBoundary-SMOTE and VIS_RST. By combining all the previous methods, the authors formulated 84 different combinations. Classification and Regression Tree (CART) was considered for comparative analysis. For training and testing, 5 stratified cross-validation was used. The classification performance was evaluated with AUC, G-Mean and F1-score. The Friedman test with Holm's post hoc test was used to compare different methods over different datasets. The results have shown that for missing data imputation with noisy imbalanced datasets, mice and k-NN performed better when the missing rate was increased. SMOTE-ENN performs similarly to SMOTE-TomekLink, SOI-CJ and SMOTE-IP in almost all noise percentage and is similar to VIS_RST with an increase in noise level. In case of incomplete and noisy imbalanced datasets, SMOTE-ENN performed well too with an increase in attribute noise percentage. The combination of MICE with SMOTE-ENN performs well when compared with other techniques.

2.5.2 Interrelation between missing and imbalanced data

This type of approaches deal with missingness and imbalance at the same time, *i.e.*, the same algorithm imputes missing values and deals with imbalance, either balancing the data or giving more importance to the minority class.

Liu et al. [46] proposed a Fuzzy-Based Information Decomposition (FID) method that addresses missing and imbalanced data simultaneously. The authors divided their work in two parts: weighting, where they determine the contribution of all observed data to estimate the missing values, and recovery, where the missing values are estimated taking into account the contribution of each observed instance. They evaluated the performance of their approach using 27 complete public datasets with a variety of characteristics. The *IR* varies from 1.05 to 42.22, the size of the datasets goes from 62 instances to 17186 and the number of features ranges from 3 to 2000. The missing values are created randomly and the missing rates they experimented are 5%, 10% and 20%. They applied 5-fold cross validation and the metrics used to measure the performance were Geometric Mean (GM), AUC and Matthews Correlation Coefficient (MCC). The authors compared the performance of the FID with other 29 approaches that combined missing values recovery methods (Mixture Kernel-Based (Mix) imputation, k-NN imputation and Self-Organizing Maps for Imputation (SOMI)) with imbalanced data learning methods (Random

Undersampling (RUS), Random Oversampling (ROS), SMOTE, Cluster Based Undersampling (CBUS), Cluster Based Oversampling (CBOS) and Majority Weighed Minority (MWM)). The classifier used to compare these approaches is the C4.5 DT. The authors concluded that the proposed method outperforms not only methods that focus only on one of the problems but also methods that solve the problems sequentially.

Shin et al. [11] proposed a method called Multiple Imputation-Based Minority Oversampling Technique (MI-MOTE), where the missing data in the majority class is imputed once with a multiple imputer and the minority class is replicated, to balance the dataset, and then the missing values are imputed with the multiple imputer used in the majority class. Because of stochasticity, the multiple imputer will impute different values in each duplicate. MI-MOTE can be applied before any classification algorithm. The authors used MICE as the imputer and set the maximum number of iterations to 10. 27 complete and imbalanced datasets with IR between [8.6, 129.5] and number of instances between [336, 145751] were used in the experiments, with 10%, 20%, 30%, 40% and 50% of missing rate created by randomly removing values. MI-MOTE was compared with five sequential approaches: no oversampling, random oversampling, SMOTE, B-SMOTE and ADASYN. The imputation is always performed with MICE. Each approach were evaluated using three classifiers: RF, Logistic Regression (LR) and Neural Network (NN). The performance of each method was measured using 5-fold cross-validation with the metrics F1-score and AUC. The results showed that MI-MOTE outperformed the baseline approaches, being more effective when the missing rate is higher. However, this method has some limitations. MI-MOTE is computationally expensive because of the multiple imputer, it would not work well when the missing rate is low because it would simply replicate minority samples and cause overfitting.

Saqib et al. [1] proposed a Conditional Generative Adversarial Imputation Network (CGAIN), which aims to impute missing values conditional to their class, taking into account their class characteristics. The generator produces faking data using the original data with missing values, class labels and random data. Then, the discriminator receives the generated data and predicts which ones were missing in the original dataset. Lastly, the generator receives the prediction performance of the discriminator and adapts his weights. Both the generator and the discriminator are fully connected NN with two hidden layers. The number of neurons in each layer are three times the number of features of the dataset. The authors used four binary and one multi-class complete datasets only with numerical features to evaluate the performance of the CGAIN. The missing data is created MCAR with percentages of missing from 5% to 20%. They compared their approach with a

Generative Adversarial Imputation Network (GAIN), only using MICE, RF and matrix completion, using the RMSE as metric. The IR of the datasets is between 1,14 and 3,52. To evaluate the performance of the CGAIN on imbalanced datasets, the authors deleted rows of one of class to create the desired imbalance. The tested IR was {9, 3, 1.5, 1}. The proposed CGAIN outperformed in every test the other approaches with the original datasets and the datasets with specific imbalance. The RMSE varies between 0,0601 and 0,2329.

2.5.3 Summary

In this section, a literature review on the topic of class imbalance and missing data was performed. There are only a few researches that deal with class imbalance and missing data simultaneously. In every work, the missing data was considered MCAR, since it is easier to deal with. The missing values was created artificially or the datasets already had already missing values. The missing rate was, in most papers, between 5% and 20%. The IR of the used datasets has a wide range. k-NN and MICE obtained better results when performing missing imputation, while SMOTE is more common to oversample the minority class. Table 2.4 summarizes the papers reviewed in the previous sections.

There are only a few authors that proposed approaches that deal with class imbalance and missing data simultaneously. It is important to study the interrelation between these two problems in order to understand them better and propose a classification or preprocessing algorithm that deal with both problems considering the other and not only solve them individually.

Paper	Algorithms			Datasets		Metrics	Compared methods	Conclusions
	Imputation	Class Imbalance	Classification	IR	Missing Rate			
[16]	NB	ADASYN	RF	1,87	<i>NS</i>	NB	NB, SVM, DT	The proposed approach outperformed the other ones, with an accuracy of 87,10%
[43]	RFR or statistical methods	Undersampling	DNN based on AutoHPO	55,3	2 features with 0,3 and 0,03	Accuracy, specificity, sensitivity, G-Mean, FNR and FPR	DNN, Bagging, RF, AdaBoost and XG-Boost	Their approach can improve the FNR Mean 1,7% higher than the mean of the compared methods
[44]	k-NN	Higher weight	Weighted k-NN	59	<i>NS</i>	5-fold cross-validation Cost function defined by IDA	SVM, RF and glsad-aboost	The proposed method outperformed the compared methods
[45]	EM, MICE, k-NN, mean and median	13 types of SMOTE	CART	[1.82, 58.40]	0%, 5%, 10%, 15% and 20%	5 stratified cross-validation AUC, G-Mean and F1-score	<i>NA</i>	MICE and k-NN obtained better results; SMOTE-ENN performs similarly to SMOTE-TomekLink, SOI-CJ and SMOTE-IP; MICE with SMOTE-ENN obtained the better results
[46]	FID		C4.5 DT	[1.05, 42.22]	5%, 10% and 20%	5-fold cross validation GM, AUC and MCC	29 combined approaches	The approach outperforms all the other approaches, not only the ones that focus on one method, but sequential approaches
[11]	MI-MOTE		RF, LR and NN	[8.6, 129.5]	10%, 20%, 30%, 40% and 50%	5-fold validation cross and AUC	MICE followed by: no oversampling, random oversampling, SMOTE, B-SMOTE and ADASYN	MI-MOTE has some limitations; this approach outperforms the others, specially when the missing rate is high
[1]	CGAIN		<i>NA</i>	[1.14; 3.52] 9, 3, 1.5, 1	5% to 20%	RMSE	GAIN, MICE, RF and matrix completion	CGAIN outperformed in every test

Table 2.4: Summary of the literature review

This page is intentionally left blank.

Chapter 3

Experiments

In Section 2.5, it can be concluded that the interrelation between missing data and distribution based irregularities has yet to be studied. These data quality problems are common in real world datasets and their relation might be important to improve the classification performance. For this reason, this thesis will focus on the interrelation between these irregularities instead of considering the two problems individually. In a preliminary stage, the main focus was to study the impact distribution based problems have on missing data imputation. This phase can be divided in two steps:

1. Class imbalance VS missing data imputation
2. Distribution based VS missing data imputation

After analysing the results obtained in the preliminary work, an approach referred in Section 2.5 was modified in order to study the effect distribution based irregularities have on classification problems when the data has missing values. In order to do so, an approach described in Section 2.5 will be modified to consider the distribution of the minority class.

In this section, the performed experiments will be described and their results will be discussed. The datasets used for the experiments are described in Section 3.1.

3.1 Datasets

Two types of datasets were collected: 150 real binary datasets from UCI [47], Kaggle [48] and OpenML [49] and artificial ones generated with a tool developed by

Wojciechowski et al. [50] to create multidimensional and multi-layer datasets. In this work, this tool will be referred as *datagenerator*. Since the latest datasets were created in a controlled environment, they were used in a preliminary stage to validate the initial assumptions and serve as base to the rest of the work, but it is important to study the problem in an uncontrolled scenario and confirm if the tendencies observed with the artificial datasets remain, that is why most of the analysis performed in this thesis uses real world datasets.

The artificial datasets were generated with 1400 examples, 3 features (x , y and z) and $IR \in \{2, 4, 6, 10, 20, 50\}$. The reason why the number of samples is 1400 is because in [50], the authors generated two datasets with 1200 and 1500 samples and 1400 is between these two values. The chosen IR values represent highly imbalanced and less imbalanced datasets. The datasets are tridimensional to allow a visualization of the original data and the imputed data. Figures 3.1-3.6 show examples of the generated datasets projected on axis xz with $IR = 2$. The second and fourth datasets are versions of the first and third ones, respectively, with the small disjuncts problem. All datasets have a certain amount of overlap between classes.

The real datasets only have numerical features and all of them are normalized. Their characteristics are described in Table 3.1 (more detailed description of their properties is in Table B.1). Some of the datasets are balanced ($IR = 1$) to be used as a comparison to imbalanced datasets.

Characteristic	Range
IR	[1, 29.5]
no. of features	[2, 310]
no. of samples	[100, 20000]

Table 3.1: Real datasets characteristics

The next section addresses the distribution of the minority class of real datasets referred in Section 2.4.2.

Distribution of Minority Samples

Each minority sample of real datasets is labelled using the method described in Section 2.4.2. The neighbourhood of each point is defined using the k-Nearest Neighbours (k-NN) with $k = 5$. Lower values of k would not distinguish well points of different types, while higher values would not follow the assumption of the locality of the method. In this thesis, the size of the neighbourhood is the same for all datasets. Since all the features are numerical, the distance metric chosen was the Euclidean Distance.

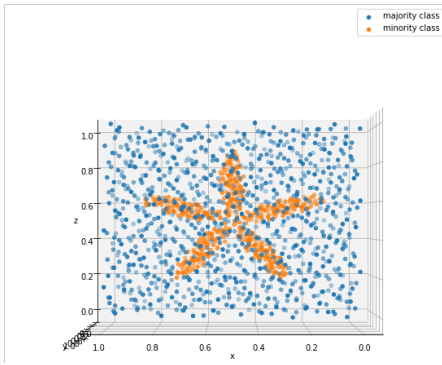


Figure 3.1: flower

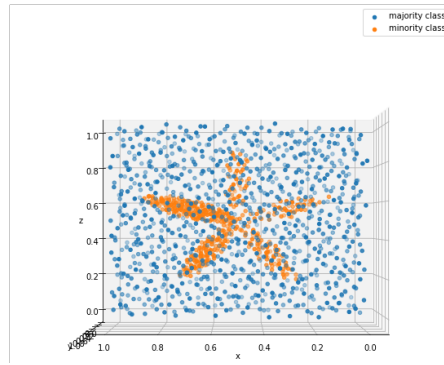


Figure 3.2: flower-min-imbalanced

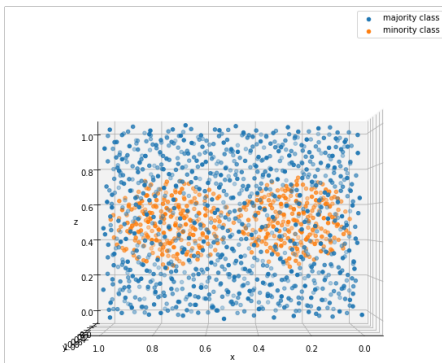


Figure 3.3: two-circle-integumental

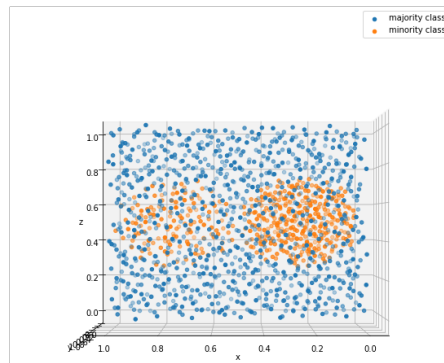


Figure 3.4: two-circle-integumental-min-imbalanced

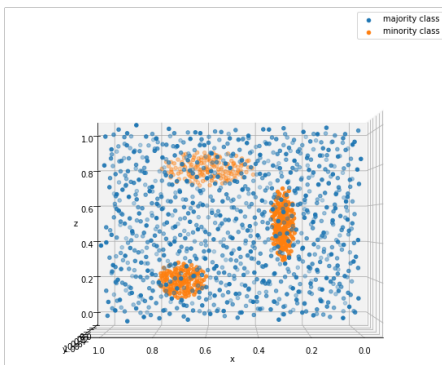


Figure 3.5: paw3

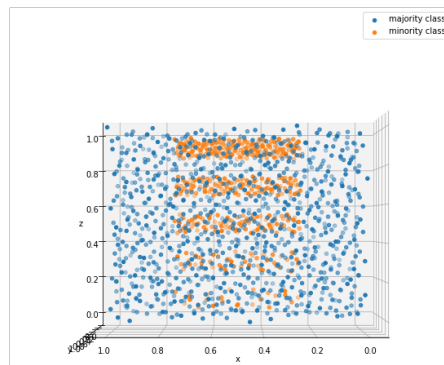


Figure 3.6: subclus5

After labelling all samples from the minority class, there will be a percentage of points safe, borderline, rare and outlier for each dataset. The distribution of each type of point is represented in Figure 3.7. It can be concluded that most of the datasets have a higher number of safe points and a lower percentage of outliers and rare samples.

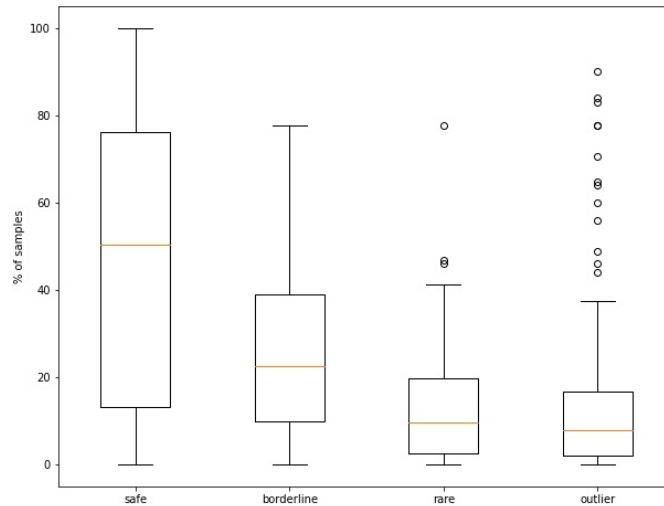


Figure 3.7: Distribution of each type of sample in real world datasets.

3.2 Preliminary Work and Assumptions

In a preliminary stage, the main goal was to find out if the IR alone had a critical role on the imputation error. According to Puri et al. [45], MICE and k-NN perform better than other baseline imputation algorithms when the classes are imbalanced. Therefore, these are the two methods used to perform the imputation of the missing values. The estimator used in MICE is a linear regression and the maximum number of iterations is 10. In k-NN, the number of neighbours k is 5. The missing mechanism considered is the MCAR since these missing values values are easier to handle. In future work, it might be interesting to consider other missing mechanisms.

The experiments for each dataset followed 3 steps:

1. Create an incomplete dataset MCAR (all the missing values will be randomly deleted). The missing values are distributed equally among all features and following the original IR. *E.g.*, consider a binary dataset with 100 samples and

4 features ($100 \times 4 = 400$ values) with $IR = 4$ (the minority class represents 20% of the samples). If the desired Missing Rate (MR) is 25%, the dataset will have $0.25 \times 400 = 100$ missing values. Each feature will have $100/4 = 25$ missing values and, to maintain the IR, the minority and majority class will have $0.2 \times 100 = 20$ and $0.8 \times 100 = 80$, respectively;

2. Impute the missing values created before;
3. Calculate the error of the imputation.

For each dataset, the previous steps are performed 10 times. The mean of the errors for each dataset is calculated and used for the analysis. The pseudocode for this approach is in Algorithm 1, where \mathcal{I} is MICE and k-NN and the Missing Rate (MR) is $\{5\%, 10\%, 20\%, 40\%\}$. Since the results for the MR 5%, 10% and 40% are similar to 20%, they are presented in Appendice D.

Algorithm 1 Imputation

Input: dataset D , missing rate mr , imputer \mathcal{I}

Output: imputation error $finalerror$

```
1: procedure
2:    $finalerror \leftarrow 0$ 
3:   for  $i=0:10$  do
4:      $D_{miss} \leftarrow$  dataset with  $mr$  of missing values created MCAR
5:      $\mathcal{I} \leftarrow$  imputer fitted on  $D_{miss}$ 
6:      $D' \leftarrow$  original samples with with missing values
7:      $D'_{miss} \leftarrow$  samples with missing values
8:      $D'_{imp} \leftarrow D'_{miss}$  with missing values imputed using  $\mathcal{I}$ 
9:      $error \leftarrow$  imputation error between  $D'$  and  $D'_{imp}$ 
10:     $finalerror \leftarrow finalerror + error$ 
11:   end for
12:    $finalerror \leftarrow \frac{finalerror}{10}$ 
13: end procedure
```

Two metrics were used to measure the imputation error: RMSE and MAE. RMSE is a widely used metric to compare imputation methods [51, 52] but might not be appropriate when the difference between the original and imputed values are high. MAE is also calculated because it is less affected by larger errors [52]. The Pearson Correlation Coefficient was also calculated but, since the results were similar, they will not be showed.

The next two sections will show the imputation results and discuss them relating the error with other types of data irregularities.

3.2.1 Missing Features and Class Imbalance



Figure 3.8: Types of data irregularities. Adapted from Das et al. [5]

This section will focus on the effect the class imbalance has on the imputation error. It is expected that when the IR increases, the imputation error will also increase, since the minority class is less represented and the missing values on minority samples will have less examples to perform the imputation.

Initially, the previous approach was used in the artificial datasets. Figures 3.9 and 3.10 presents the results obtained using k-NN and MICE, respectively, with 20% of missing values. The results obtained for the other percentages of missing are shown in Appendice C. Since the results are similar with 20% os missing values, only this percentage will be evaluated in this section. It can be concluded that with the increasing of the IR, the imputation error will also increase. This increasing follows something similar to a logarithmic function. For IR higher than 10, the imputation error increases slowly. MICE obtained lower errors than k-NN, but the increasing

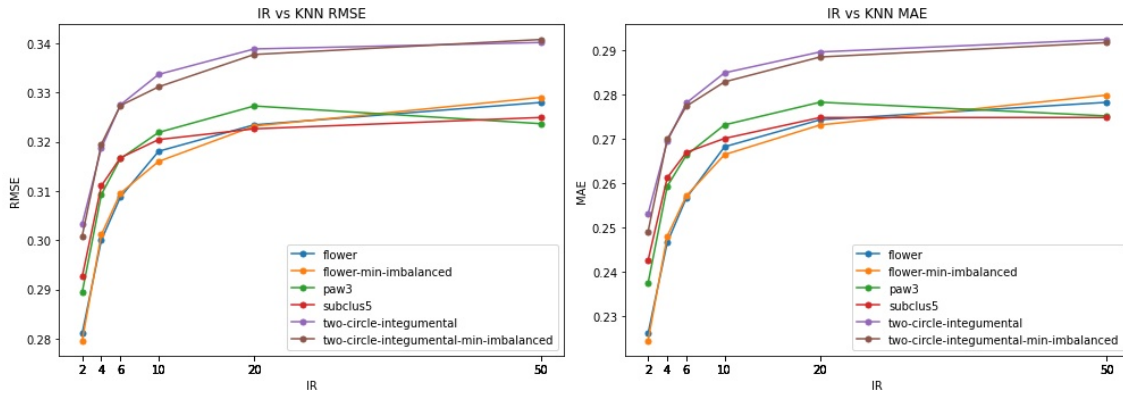


Figure 3.9: Imputation error of the artificial datasets using k-NN with 20% of missing values.

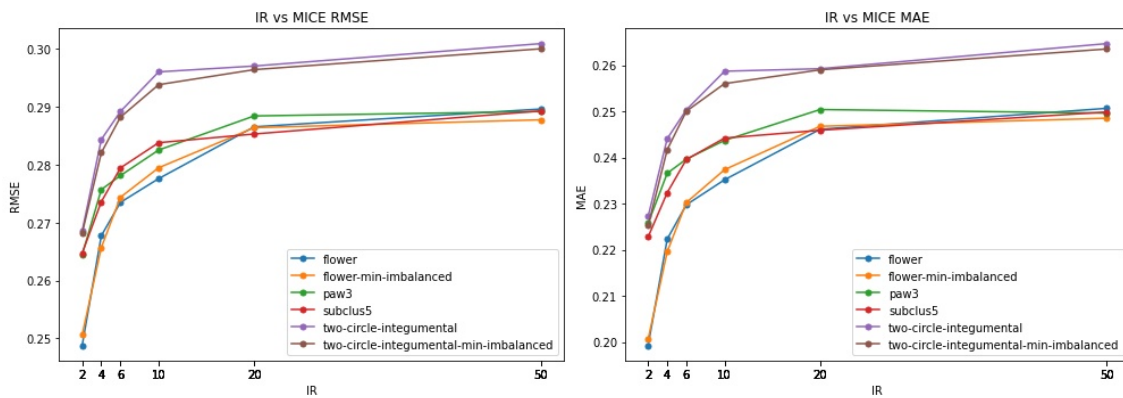


Figure 3.10: Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 20% of missing values.

is similar on both approaches. The two datasets with the small disjuncts problem (*flower-min-imbalanced* and *two-circle-integumental-min-imbalanced*) obtained similar results as their version without the imbalance in the minority class.

Since these results are obtained in a controlled scenario, where the data follows a normal distribution, the missing imputation errors are the expected ones. Most real world datasets almost never follow a normal distribution, therefore, it is important to perform this analysis on these type of data to validate the results obtained on artificial datasets.

Figures 3.11 and 3.12 show the RMSE and MAE of the imputation using k-NN and MICE, respectively. It can be observed that datasets highly imbalanced have a high imputation error, therefore, the initial assumption, where it is expected that the error will increase with the IR, is verified, but some datasets more balanced (lower IR) also have a high imputation error, which is not in agreement with the assumption.

In order to try to explain better the imputation error, the analysis will focus on the

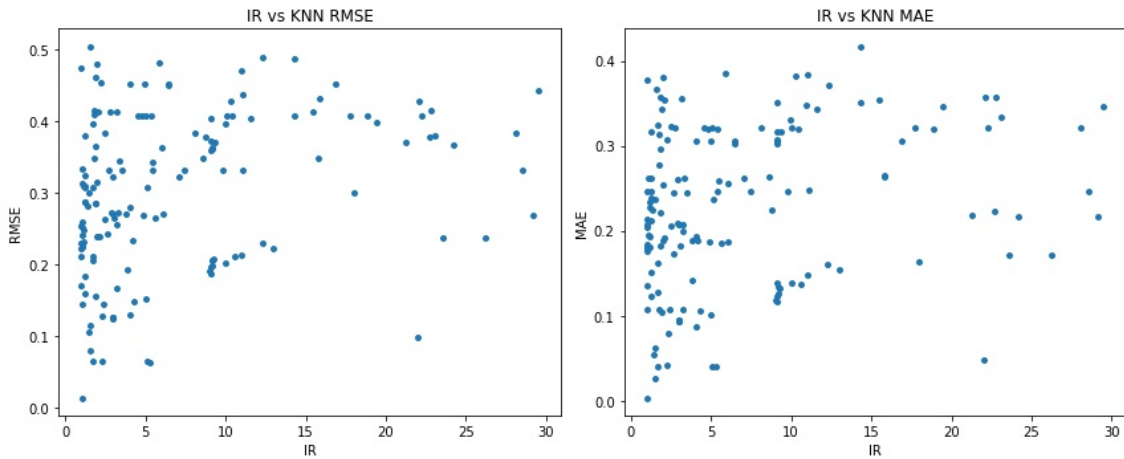


Figure 3.11: Imputation error using k-NN of datasets with different IR and 20% of missing values.

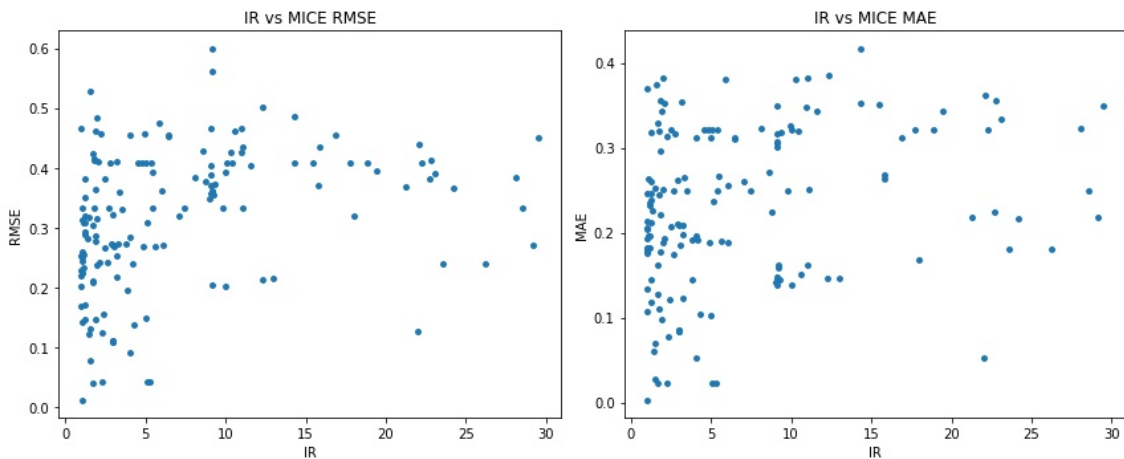


Figure 3.12: Imputation error using MICE (with Linear Regression as estimator) of datasets with different IR and 20% of missing values.

the relation between it and some of the most simple characteristics of the datasets. These characteristics are the number of samples in the dataset (*size*), the number of features (*n_features*) and the IR. The ρ (Section 2.3.1) is good to measure the association between two variables [34]. Positive values of ρ indicate that when one of the two variables increases, the other also increases. If ρ is negative, when one of the variables increases, the other decreases. The higher the values of $|\rho|$, the more correlated the variables are.

The results of this analysis are shown in Figure 3.13. Higher values of $|\rho|$, *i.e.*, more positive or more negative, correspond to darker blue and darker red cells, respectively, which means a higher correlation. The IR is the characteristic with a higher correlation to the imputation error, especially to the RMSE when the imputer is MICE, although it is not very high. For this experimental setup, the size of a dataset did not present as a critical feature for the imputation. The number of

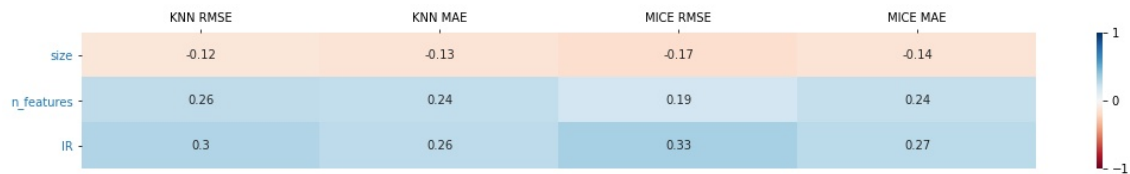


Figure 3.13: Pearson Correlation Coefficient between the imputation error and some datasets' characteristics.

features has a low positive correlation with the error, which means that when the number of features is higher, the imputation error will slightly increase.

In this section, it can be concluded that:

- The results for the artificial datasets confirm the original assumption that the increasing of the IR affects negatively the imputation of the missing values;
- The imputation error is high for real world datasets with high IR;
- Balanced real world datasets have both high and low imputation error, which does not agree with the initial assumption.

The unexpected results for balanced datasets with a high imputation error led to the next sections, where other characteristics of the datasets will be studied. From now on, the analysis of the imputation error will only focus on real datasets since for the artificial ones obtained the intended results.

3.2.2 Missing Features and Complexity Metrics

The results for highly imbalanced datasets obtained in the previous section were expected but, for datasets with a low IR, the imputation error should be also low, since both classes are similarly represented. In an effort to explain the unexpected errors, it was tried to find a relation between the complexity of a dataset and the imputation error. It is expected that the more complex a dataset is, *i.e.*, the higher the metrics' values described in Section 2.4.1, the higher the imputation. These metrics were calculated using the *pymfe* [53] package from Python. This package computes all measures defined by Lorena et al. [39].

In Figure 3.14, it can be seen the correlation between the complexity metrics and the imputation error using each approach. In this case, the two features in the ρ are one of the complexity metrics and the imputation error metric using one of the approaches. All metrics are divided in four main groups: network, feature-based,

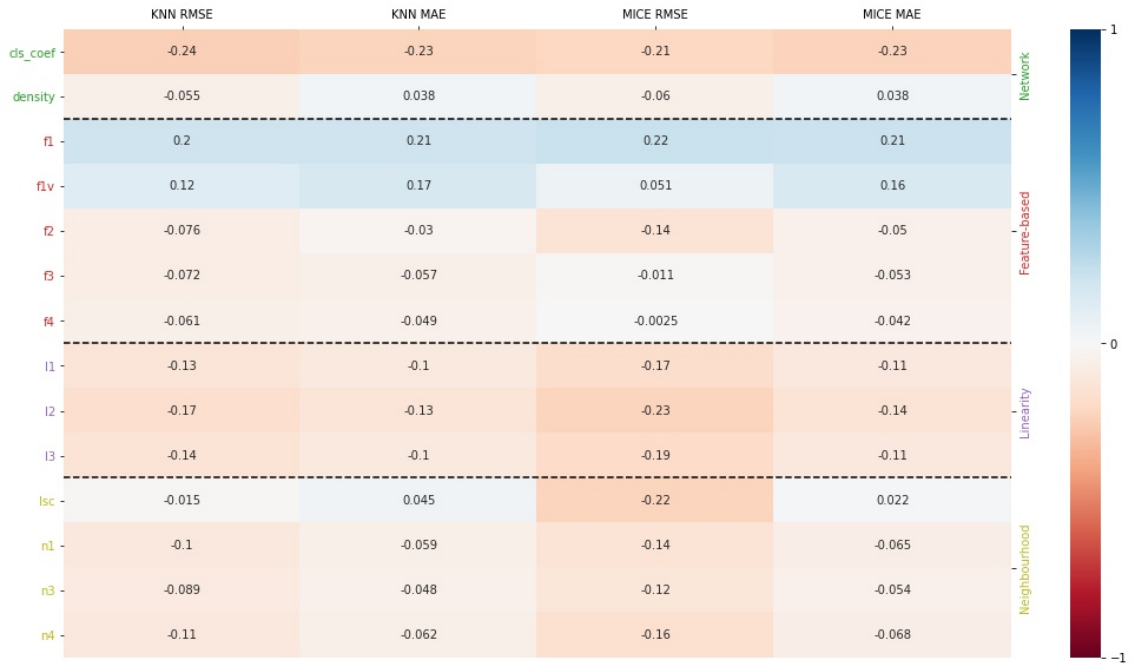


Figure 3.14: Pearson Correlation Coefficient between the imputation error and the complexity metrics.

linearity and neighbourhood. In general, the correlation values are low and negative (less complex dataset, higher imputation error), which was not expected. The results claim that the more complex a dataset is, the higher the imputation error will be. The only two metrics that obtained the opposite were $f1$ and $f2$.

In order to explain unexpected results, the datasets that obtained a higher MAE than the Third Quartile (3Q), *i.e.*, the 25% of datasets with a higher imputation error, were analysed. The mean size of the dataset was the only characteristic that deviated from the original set of datasets. The mean size of all datasets was 1557 while the datasets with a error higher than 3Q had a mean size of 420. The standard deviation was 3118 and 232, respectively. It means that datasets with a higher number of samples have a lower error imputation.

The correlation absolute values are small, which means that there is almost no correlation between the imputation errors and the complexity metrics.

3.2.3 Missing Features and Distribution Based



Figure 3.15: Types of data irregularities. Adapted from Das et al. [5]

The conclusions of the correlation between the complexity metrics and the imputation error were not the expected ones. For that reason, other characteristics of the datasets were analysed to explain the imputation error obtained: the distribution of the minority class. The percentage of each type of sample (safe, borderline, rare and outlier) is calculated using the approach described in Section 3.1.

The analysis performed in this section is similar to the previous one with the complexity metrics. It is expected that datasets with a higher percentage of safe points, *i.e.*, with a higher number of samples with a similar neighbourhood, will have a lower imputation error because each sample will have more similar examples closer to them to predict the missing values. Datasets with a higher number of unsafe samples, specially rare and outliers, will have a higher error.

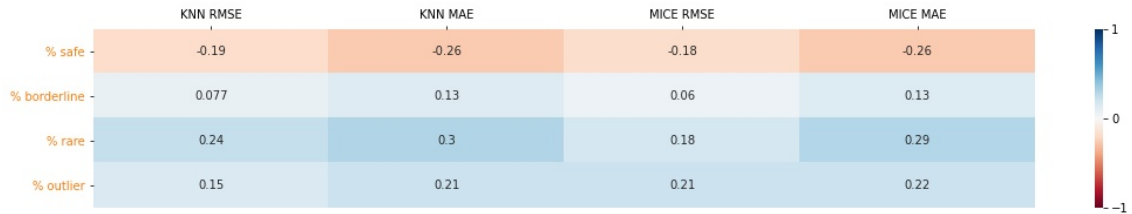


Figure 3.16: Pearson Correlation Coefficient between the imputation error and the distribution of the minority class.

Figure 3.16 represents the correlation between the percentage of each type of point and the imputation error using k-NN and MICE. As expected, the higher the percentage of safe examples, the lower the imputation error (low and negative correlation values). The higher the percentage of rare and outliers, the higher the error (high and positive correlation). Borderline samples do not have a high impact on the imputation, since they are examples that can have more samples of the same or the opposite class in the neighbourhood.

3.2.4 Main conclusions

After the performed analysis in this section, it can be concluded that:

- The IR and number of features have some impact on the imputation error: the higher the imbalance and the number of features, the higher the error;
- Complexity metrics have a low impact on the imputation (low correlation values). The results were not the anticipated ones, since it was expected that more complex datasets (higher metrics values) would have a higher imputation error;
- The distribution of the minority class have a slightly high impact on the imputation: datasets with a higher percentage of safe examples and lower of unsafe samples will have a lower imputation error.

The analysis performed until now only considered the imputation error but, once the missing data is imputed, it is important to evaluate the performance of the classification [54]. A dataset with a high imputation error might have a high classification performance because a sample can have missing values that are predicted wrong but closer to values from samples from the same class, therefore, its label will be predicted correctly.

For this reason, the analysis will now focus on the classification performance on an imbalanced scenario with missing values.

3.3 Data irregularities and classification task

In Section 2.5, some approaches that deal with missing data and class imbalance simultaneously were reviewed. FID presented some limitations when trying to consider only some types of minority samples when oversampling the minority class and CGAIN revealed some problems when used on the collected datasets. Therefore, MI-MOTE was chosen to be modified to take into consideration the distribution of the minority class.

Shin et al. proposed MI-MOTE [11] as a preprocessing algorithm to impute missing data and, at the same time, oversample the minority class to solve the imbalance problem. Its pseudocode is in Algorithm 2, where $x_i = (x_{i1}, \dots, x_{id}) \in \mathbb{R}^d$ is an input vector, y_i is the corresponding class label for the i -th instance, $D = (x_i, y_i)_{i=1}^N$ is a training dataset and \mathcal{M} is a multiple imputer fitted with the original training dataset. Missing values in majority samples are simply imputed using \mathcal{M} while the minority class is oversampled first and then imputed with \mathcal{M} . Minority samples that have missing values are replicated λ times (if $\lambda = 5$, there will be 6 replicas for each oversampled example) and then all the minority class is imputed using \mathcal{M} . Because of stochasticity, replicas will be imputed with different values. In the experiments, λ is calculated previously so that the IR is approximately 1, *i.e.*, so both classes have the same number of samples. *E.g.*, if the majority and minority classes have 300 and 200 examples, respectively, and the minority class has 20 samples with at least one missing value, λ will be 5 (the minority class will now have $200 + 5 \times 20 = 300$ samples).

This approach only takes into account the IR, ignoring other data distribution based irregularities. The main goal in this section is to inquire if the distribution of the minority class has an impact not only on the missing data imputation but also on the classification task, since, in most cases, this is the end goal. In order to do so, MI-MOTE was modified to consider the minority class distribution.

Consider t_i the type of the i -th instance of D , where $t_i \in \{m, s, b, r, o\}$ and m corresponds to the majority samples, s , b and r are safe, borderline and rare examples, respectively, and o are the outliers. In this improved method, when performing the oversampling, only minority examples from specific types will be replicated. Algorithm 3 shows the changes made to the original approach.

The main problem at this point was to select the types of points to oversample. Sáez et al. [42] studied the effect different types of minority samples have on the classification process, so their work will be used as a starting point to decrease the amount of experiments in this section. The authors concluded that oversampling

Algorithm 2 MI-MOTE

Input: training dataset $D = (x_i, y_i)_{i=1}^N$, oversampling ratio λ **Output:** multiple imputer \mathcal{M} , classifier f

- 1: **procedure**
 - 2: $\mathcal{M} \leftarrow$ multiple imputer fitted on D
 - 3: \triangleright *majority instances*
 - 4: $D_0 \leftarrow \{(x_i, y_i) \mid (x_i, y_i) \in D \text{ and } y_i = 0\}$
 - 5: $\hat{x}_i \leftarrow \mathcal{M}(x_i)$ for $\forall (x_i, y_i) \in D_0$
 - 6: $D'_0 \leftarrow \{(\hat{x}_i, y_i) \mid (x_i, y_i) \in D_0\}$
 - 7: \triangleright *minority instances*
 - 8: $D_1 \leftarrow \{(x_i, y_i) \mid (x_i, y_i) \in D \text{ and } y_i = 1\}$
 - 9: $x_i^{(0)}, \dots, x_i^{(\lambda)} \leftarrow$ copies of x_i for $\forall (x_i, y_i) \in D_1$
 - 10: $\hat{x}_i^{(l)} \leftarrow \mathcal{M}(x_i^{(l)})$ for $\forall (x_i, y_i) \in D_1$ and $l = 0, \dots, \lambda$
 - 11: $D'_1 \leftarrow \{(\hat{x}_i^{(l)}, y_i) \mid (x_i, y_i) \in D_1 \text{ and } l = 0, \dots, \lambda\}$
 - 12: \triangleright *refined training dataset*
 - 13: $D' \leftarrow D'_0 \cup D'_1$
 - 14: Train a classifier f with D'
 - 15: **end procedure**
-

Algorithm 3 Modified MI-MOTE

Input: training dataset $D = (x_i, y_i)_{i=1}^N$, oversampling ratio λ , types of training samples $t = (t_i)_{i=1}^N$, types to replicate r **Output:** multiple imputer \mathcal{M} , classifier f

- 1: **procedure**
 - 2: $\mathcal{M} \leftarrow$ multiple imputer fitted on D
 - 3: \triangleright *majority instances*
 - 4: $D_0 \leftarrow \{(x_i, y_i) \mid (x_i, y_i) \in D \text{ and } y_i = 0\}$
 - 5: $\hat{x}_i \leftarrow \mathcal{M}(x_i)$ for $\forall (x_i, y_i) \in D_0$
 - 6: $D'_0 \leftarrow \{(\hat{x}_i, y_i) \mid (x_i, y_i) \in D_0\}$
 - 7: \triangleright *minority instances*
 - 8: $D_1 \leftarrow \{(x_i, y_i) \mid (x_i, y_i) \in D \text{ and } y_i = 1\}$
 - 9: $x_i^{(0)}, \dots, x_i^{(\lambda)} \leftarrow$ replicated copies of x_i for $\forall (x_i, y_i) \in D_1$ and $t_i \in r$
 - 10: $\hat{x}_i^{(l)} \leftarrow \mathcal{M}(x_i^{(l)})$ for $\forall (x_i, y_i) \in D_1$ and $l = 0, \dots, \lambda$
 - 11: $D'_1 \leftarrow \{(\hat{x}_i^{(l)}, y_i) \mid (x_i, y_i) \in D_1 \text{ and } l = 0, \dots, \lambda\}$
 - 12: \triangleright *refined training dataset*
 - 13: $D' \leftarrow D'_0 \cup D'_1$
 - 14: Train a classifier f with D'
 - 15: **end procedure**
-

certain types of examples can increase the classification performance. They tested all possible combinations of types and the results showed that the best configuration for each dataset usually preprocessed borderline samples, for this reason, this type will be replicated in every experiment on this theses ($r = \{b\}$). In addition, the authors concluded that outliers should be oversampled if the percentage of safe samples is low. Therefore, these two types are combined with borderline ones ($r = \{b, s\}$, $t = \{b, o\}$, $r = \{b, s, o\}$). The last combination has a particularity, r will be $\{b, s\}$ if the percentage of safe examples is higher than outliers and r is $\{b, o\}$ if the dataset has more minority samples outliers than safe. That means that r will depend on the percentage of safe and outliers of each dataset.

The results are compared with the original MI-MOTE, when all types are preprocessed, and a baseline approach, where the missing data is first imputed using k-NN and then the minority class of the trainset is oversampled using SMOTE-ENN (these choices were based on the results obtained by Puri et al. [45] referred in Section 2.5). The comparison is performed using a RF f implemented using the *scikit-learn* package with the default parameters. The performance of the classifier was evaluated through stratified cross-validation and the metric used was the F1-score. For each complete dataset, the missing data was first created MCAR the same way explained in Section 3.2. In most works mentioned in Section 2.5, the missing rate is between 5% and 50%. In this experiments, only 20% of MR will be discussed, because MI-MOTE performs better when the missing rate is high but the baseline approach is better for lower percentages of missingness. The results for the remain percentage of missingness are in Appendice D. They are similar to the 20% of missingness, therefore, it is not necessary to study them separately. In both approaches, each dataset was divided into 5 folds where each fold was used once as the testset and the others as the trainset.

In the baseline approach, the missing values are first imputed and then the minority class is oversampled. After training the RF with each oversampled trainset, the labels of the corresponding testset are predicted and the F1-score is calculated. This process is repeated 10 times and the mean of the F1-scores is stored.

On the experiments using MI-MOTE, the algorithm is used in the preprocessing phase instead of imputing and then oversampling. This process is also repeated 10 times and the mean of the F1-scores is stored.

In the end, there will be six F1-scores for each dataset: *baseline*, *MI-MOTE*, *MI-MOTE b*, *MI-MOTE b+s*, *MI-MOTE b+o*, *MI-MOTE b+s/o*. If it is not possible to oversample the desired types of samples, the F1-score for this dataset will be *Nan*, e.g., if a dataset does not have borderline samples, *MI-MOTE b* will be *Nan*. For

this reason, the total number of datasets for each approach will be different. Table 3.2 presents the mean of the F1-scores obtained for all datasets (first row) and a comparison between all approaches. The main goal is to prove that the approaches in each column improve the classification performance compared to the baseline and the original MI-MOTE. It is pointless to compare baseline with MI-MOTE twice, therefore, the baseline column only has the mean row filled in. Consider *base* the row approach and *compare* the column one.

For each comparison, it is shown the number and the percentage of datasets (# and %, respectively) that obtained a higher F1-score using *compare*. Additionally, the mean difference between the results of *compare* and *base* (*diff*) is also shown. Higher values of *diff* implies that *compare* further improves the classification performance. For example, reading the comparison between MI-MOTE and the baseline approach, 119 datasets out of 150 had a higher F1-score when using MI-MOTE to process the data than using baseline. It corresponds to 79.33% of the datasets. On average, the F1-score using MI-MOTE is 0.0443 higher.

		Baseline	MI-MOTE	Modified MI-MOTE			
				<i>b</i>	<i>b + s</i>	<i>b + o</i>	<i>b + s/o</i>
mean		0.713	0.757	0.745	<u>0.770</u>	0.746	0.788
baseline	#	-	119 (150)	118 (141)	<u>135</u> (150)	119 (142)	136 (150)
	%	-	79.33%	83.69%	<u>90.0%</u>	83.8%	90.67%
	diff	-	0.0443	0.0428	<u>0.0568</u>	0.0464	0.0756
MI-MOTE	#	-	-	86 (141)	<u>118</u> (150)	87 (142)	125 (150)
	%	-	-	60.99%	<u>78.67%</u>	61.27%	83.33%
	diff	-	-	0.0006	<u>0.0125</u>	0.0014	0.0313

Table 3.2: F1-score results of baseline, MI-MOTE and modified MI-MOTE. The best and second best results are in bold and underlined, respectively.

Looking into Table 3.2, as concluded by Shin et al. [11], MI-MOTE improves other baseline methods. Additionally, it can be concluded that *MI-MOTE b+s/o* got the better results, with a mean F1-score of 0.788. *MI-MOTE b+s* is a close second with a mean F1-score of 0.770. The results show that considering the minority class distribution can improve the classification performance when certain types of samples are chosen. Oversample only borderline examples (*MI-MOTE b*) or borderline and outliers (*MI-MOTE b+o*) did not show much improvement over the original approach.

Figure 3.17 shows the correlation between some characteristics of the datasets (rows)

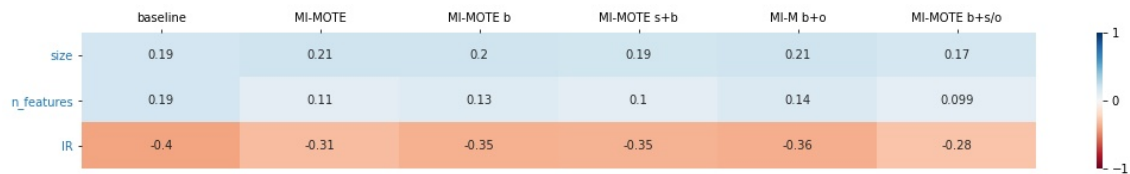


Figure 3.17: Pearson Correlation Coefficient between the F1-score and the datasets characteristics.

and the F1-score obtained for each dataset with all approaches (columns). Interpreting the first two rows, the number of samples and features do not have a high impact on the performance of the classification, since the correlation is considerably low. The IR (third row) has a higher impact, particularly on the baseline approach. The negative values mean that when the IR increases, the performance of the classification decreases, which is expected since the minority class is less represented and will be more difficult to recognize by the RF.

Since the results cannot be explained only using simpler datasets characteristics, other analysis will be performed. It is expected that the performance of the algorithms will decrease with the increasing of the complexity of a dataset, *i.e.*, the higher the values of the complexity metrics, the lower the F1-score. It means that the correlation between the metrics and the F1-score using each approach will be lower. Figure 3.18 shows the correlation between each complexity metric and the F1-score using each approach. In general, the results are not that significant. The correlation is, in fact, negative (more complex datasets have a lower F1-score) but the absolute value is not high enough to be significative, except the metrics that belong to the network category (that are positive but almost zero) and the $f1$ and $f1v$ metrics. When these last two metrics have a lower value, the dataset has a higher separation between classes. When two classes are more distant, there is a higher probability that the samples will be correctly classified, therefore, the F1-score will be higher. This explains the negative correlation (low $f1v$ \rightarrow high separation \rightarrow high F1-score). The remain metrics have a low implication on the performance of the RF

When considering the distribution of the minority class, it is expected that the F1-score has a high dependency on the distribution of the minority class. A dataset with a higher percentage of safe points will have a higher classification performance since the neighbourhood of each point is similar to the point itself. Rare examples and outliers are mainly surrounded by samples from the other class, therefore, there's a high probability that these examples will be misclassified.

Figure 3.19 presents the correlation between the performance of each approach and

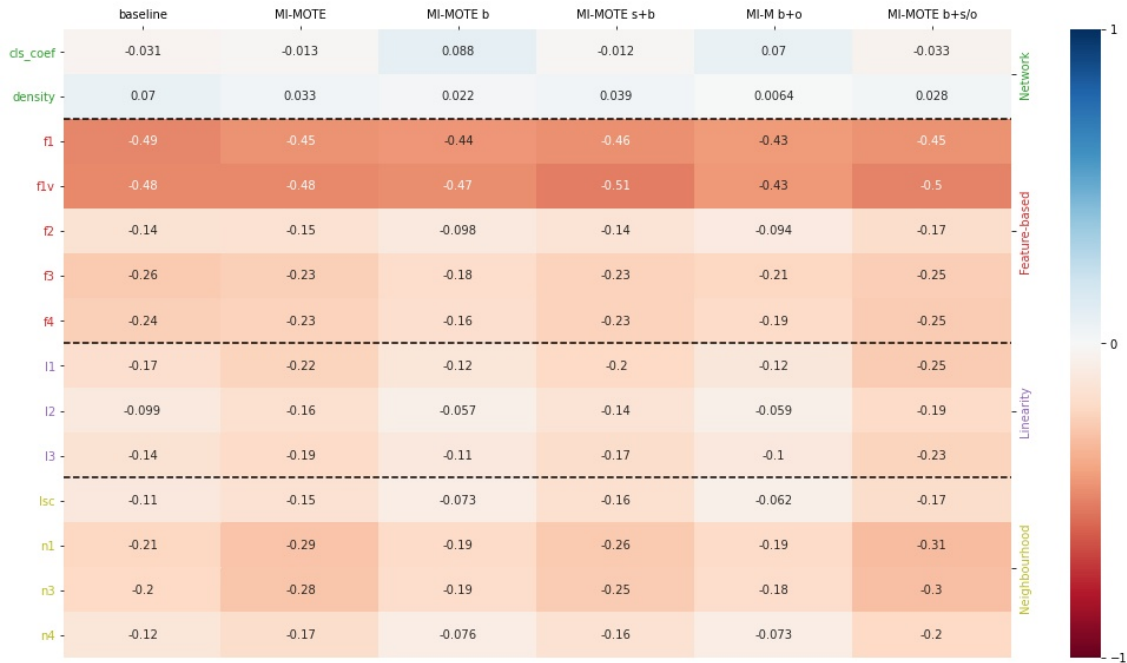


Figure 3.18: Pearson Correlation Coefficient between the F1-score and the complexity metrics.

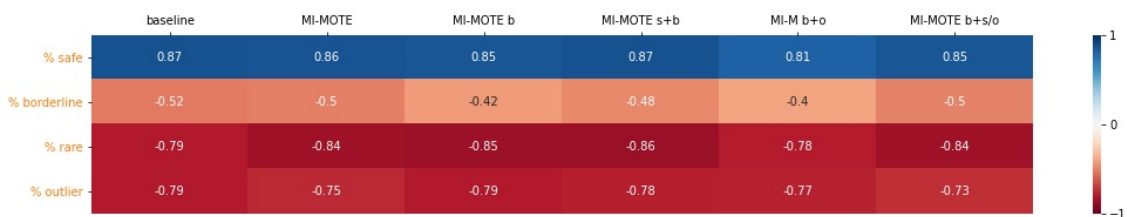


Figure 3.19: Pearson Correlation Coefficient between the F1-score and the distribution of the minority class.

the percentage of each type of minority samples. As expected, the typology of the minority class has a high impact on the F1-score. Datasets with a high percentage of safe examples (first row) are better classified than the ones with a higher percentage of unsafe examples (borderline, rare and outliers). The amount of borderline samples does not have such a high impact on the classification since the neighbourhood of a borderline sample has a mixture of samples from the same class and from the other, therefore, it can be either correctly classified or misclassified. The higher the percentage of outliers or rare samples, the lower the F1-score. It can then be concluded that the distribution (typology) of the minority class has a high impact on the classification problem. Datasets with a higher amount of safe samples are better classified than the ones with more unsafe examples.

The selection of which types of examples to oversample in MI-MOTE is important. Table 3.2 only shows the results when the configuration of the parameter r is the same for all datasets, but each dataset has different characteristics that will de-

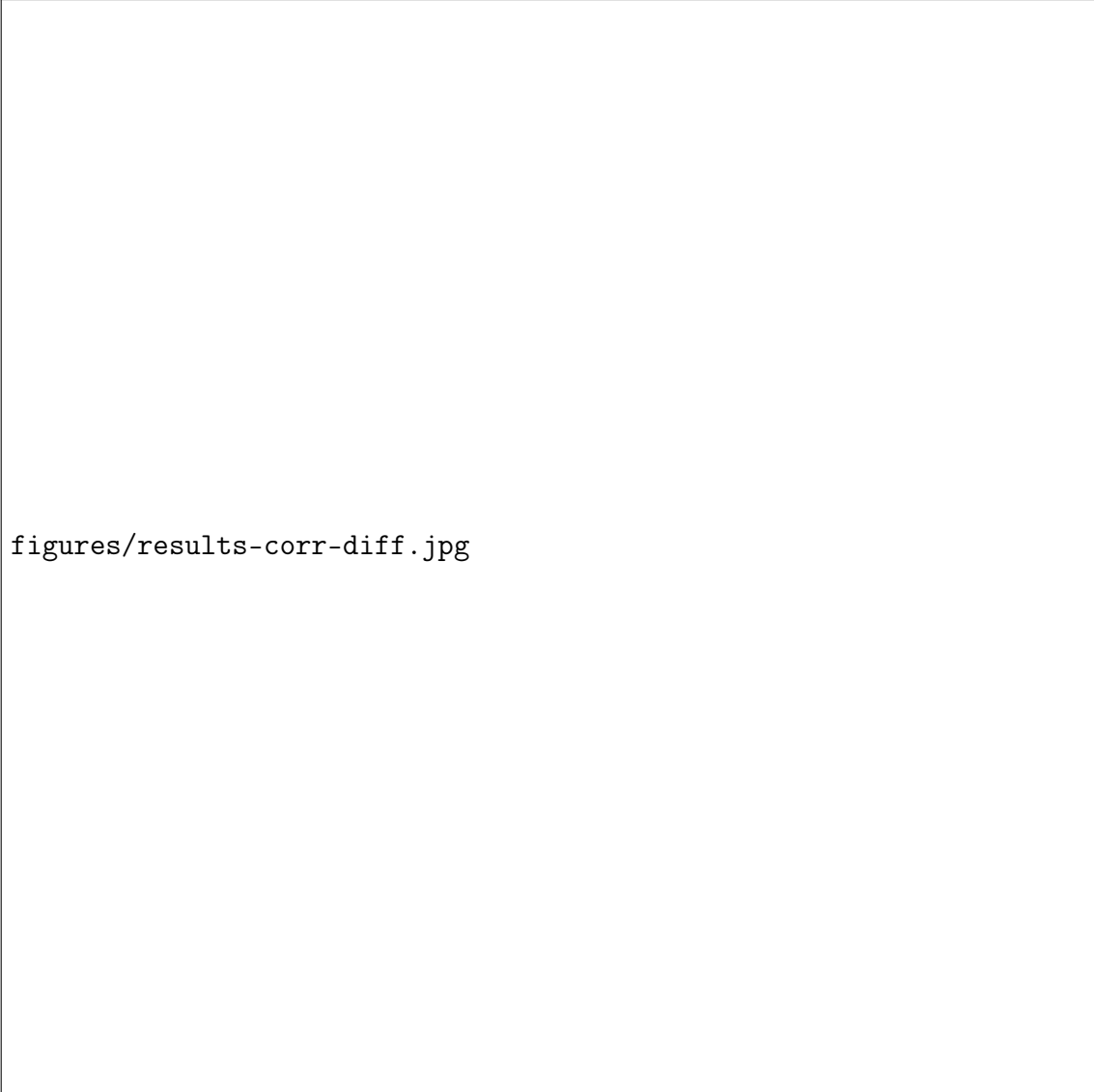
termine the best configuration for r . Considering the best configuration for each dataset (highest F1-score of the modified MI-MOTE approaches) instead oversample the same types in all datasets, 146 out of 150 datasets obtained a better result than the baseline approach (97.33%) and 145 out of 150 datasets obtained higher results when compared with the original MI-MOTE (96.67%).

In conclusion, the complexity metrics do not have a high impact on the classification performance, while the distribution of the minority class is an important analysis to explain the obtained results. That being said, in the next topic only the IR and the percentage of each type of sample will be considered. Also, the parametrization of r is important and should be chosen differently for each dataset.

As said before, *MI-MOTE b+s/o* and *MI-MOTE b+s* are a good replacement for the original MI-MOTE and the baseline approach, but in which cases are these configurations better suited? Consider *diff* the difference between the improved method and the simpler one. This value can be negative or positive, when the first approach obtained a worst or better result than the second, respectively. The higher the difference, the better the first approach. Figure 3.20 shows the correlation between *diff* and some datasets characteristics: the IR and the percentage of each type of minority sample. It is shown the results for five *diff*'s: *diff(MI-MOTE, baseline)*, *diff(MI-MOTE b+s, baseline)*, *diff(MI-MOTE b+s/o, baseline)*, *diff(MI-MOTE b+s, MI-MOTE)*, *diff(MI-MOTE b+s/o)*. The other two settings (b and $b+o$) are not presented because the results were considerably worse. It is expected that the modified MI-MOTE will improve the baseline and the original MI-MOTE when the datasets are more complex, *i.e.*, when the dataset has a higher percentage of unsafe samples, particularly rare and outliers. The higher the correlation, the higher the improvement. When the safe samples are considered, the correlation should be the opposite: the modification of the MI-MOTE should improve the original approach for lower percentages of safe samples, therefore, the correlation is negative.

In Figure 3.20, it can be seen that *MI-MOTE* improved *baseline* for higher IR and percentage of outliers. *MI-MOTE b+s* mainly increased the F1-score for higher IR when comparing the results with *baseline* and *MI-MOTE*. In fact, for higher percentages of unsafe samples, the last two approaches performed better. *MI-MOTE b+s* improved these approaches when the percentage of safe samples is high. *MI-MOTE b+s/o* produced the better results, improving *baseline* and *MI-MOTE* when both the IR and percentage of unsafe examples are high, which was the main goal.

After the classification results obtained, some main conclusions were drawn:



figures/results-corr-diff.jpg

Figure 3.20: Correlation between *diff* and some datasets characteristics.

- Using *MI-MOTE b+s/o* in the preprocessing phase improved significantly the classification performance and obtained the better results;
- The more separate the classes are, the more accurate the classification will be (*f1* and *f1v*);
- The percentage of each type of minority sample has a high impact on the classification: the higher the percentage of unsafe points, the worse the classification will be;
- *MI-MOTE b+s/o* improved the classification performance when both the IR and the percentage of unsafe samples are high, which was the main goal since *baseline* and *MI-MOTE* already deal well with balanced datasets and high percentages of safe examples.

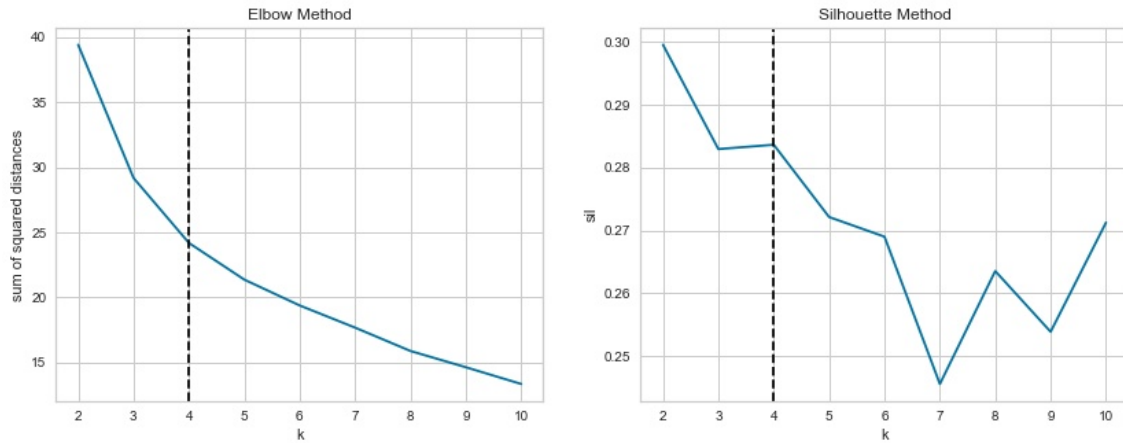


Figure 3.21: Sum of squared distances from each point to its assigned center for $k \in [2, 10]$.

3.4 Final conclusions

The previous sections performed two separate analysis: missing data imputation error and classification performance, both on imbalanced scenarios. It was concluded that the imputation error has a low correlation with the studied datasets characteristics while the classification performance has a higher correlation with said characteristics, specially with the distribution of the minority class.

While it is important to study these two topics separately, is there any connection between them? Does a dataset with a high imputation error have a low F1-score value?

In order to answer these questions, the datasets were clustered taking into account their complexity metrics. The chosen clustering algorithm was K-means because it is one of the most powerful and popular data mining algorithms [55]. The k value was chosen using two methods: elbow and silhouette methods. The first one corresponds to the sum of the squared distances from each point to its assigned center. The second method measures the quality of a clustering, *i.e.*, determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering.

Figure 3.21 shows the value of each metric for k between 2 and 10, inclusive. the results show that 4 is the best value for k . In the silhouette metric, $k = 2$ obtained the higher value but, when analysing the two clusters, the size of one of the clusters was 32 while the other was 118. Therefore, the chosen k was 4 and the sizes of the clusters are: 36, 39, 27 and 48.

Tables 3.3-3.6 show the mean values of the datasets characteristics, complexity met-

rics, distribution of the minority class and performance of the imputation and classification. In the first three tables, the cluster with the lower complexity in each column is highlighted in green while the one with the more complex datasets is highlighted in dark orange.

Table 3.4 presents the mean of each group of complexity metrics. These values are calculated using the Euclidean distance. For each dataset, it is calculated how complex the dataset is when comparing with the simplest hypothetical dataset, *i.e.*, the lower possible value of complexity for a dataset is 0, therefore, the distance between this hypothetical dataset's metrics (which are all 0) and the dataset in question's metrics is calculated using the Euclidean distance. In the column *Complexity*, all metrics are considered instead of dividing in groups of metrics. The higher these values are, the more complex the datasets in the concerned cluster are.

Cluster 2 is the one with the higher complexity in most columns, except in the IR, number of features and percentage of outliers. The mean imputation error in this cluster is the lowest in both approaches but, on the other hand, the performance of the classification is the lowest in two of the approaches and a close second lowest on the other two. That being said, it can be concluded that when the imputation error is low, it doesn't necessarily mean that the classification performance will be good. The complexity of the dataset, specially the percentage of each type of point, is important to consider when studying the classification performance. Cluster 0 obtained the lowest values in almost all columns and its classification performance is the better of all four clusters.

Some main conclusions can be drawn from the work performed on this thesis:

- While the imputation error does not have a high correlation with the studied datasets' characteristics, the classification performance does. It means that a certain missing value might be predicted far from the original one but near other samples from the same class;
- Less complex datasets have a higher classification performance (Tables 3.3-3.6);
- The imbalance ratio alone is not critical for the classification but when combined with other data quality problems is. The cluster with the more balanced datasets (cluster 2 in Table 3.3) have the highest complexity (Tables 3.4 and 3.5) and the lowest F1-score (Table 3.6).

cluster	cluster size	n_features	size	IR
0	36	11.889	643.417	5.987
1	39	50.795	1024.949	10.087
2	27	22.519	2881.148	1.798
3	48	15.229	1928.271	8.487

Table 3.3: Mean values of the datasets characteristics for each cluster.

cluster	Network	Feature	Linearity	Neighbourhood	Complexity
0	1.217	0.951	0.086	0.792	1.750
1	1.200	1.520	0.212	1.004	2.200
2	1.291	1.845	0.459	1.202	2.598
3	1.226	1.486	0.125	0.865	2.125

Table 3.4: Mean values of the complexity metrics for each cluster.

cluster	% safe	% borderline	% rare	% outlier
0	75.670	15.622	4.651	4.057
1	46.932	24.767	13.873	14.428
2	42.327	36.015	14.158	7.499
3	61.877	21.153	7.139	9.831

Table 3.5: Mean values of the percentage of each type of point for each cluster.

This page is intentionally left blank.

cluster	KNN	MICE	Baseline	MI-MOTE	b+s	b+s/o
0	0.210	0.209	0.826	0.862	0.883	0.896
1	0.263	0.264	0.644	0.684	0.676	0.714
2	0.206	0.205	0.649	0.682	0.697	0.702
3	0.209	0.213	0.720	0.780	0.801	0.816

Table 3.6: Mean imputation error and F1-score for each approach and cluster.

Chapter 4

Conclusion and Future Work

The Preprocessing phase is the one that takes more time in most data mining processes. Data quality problems (that are solved in this phase) affect negatively the classification performance. These data quality issues can be divided in two groups: distribution-based, where class imbalance and small disjuncts are included, and feature-based, where missing data belongs. These three problems often occur together in a large number of datasets. Sometimes, each problem alone are not critical for the classification task but, when combined with the others, can decrease significantly its performance.

While the interrelation among distribution-based problems have already been studied, the connection between the two types of problems have yet to be analysed. Since this topic have yet to be approached, this thesis is a study on how these two types of problems affect each other and the overall classification task.

In a preliminary stage, the imputation performance is explained using some datasets' characteristics, such as the size of the dataset, number of features and IR. It is expected that the imbalanced a dataset is, the higher the imputation error will be. It is confirmed for highly imbalanced datasets but some balanced datasets also have a high imputation error. These unexpected results led to the study of the complexity of the datasets and the distribution of the minority class. The results showed that the imputation error does not have a high correlation with the considered datasets' characteristics. In this analysis, only the imputation error was considered but a high imputation error does not mean a low classification performance.

After studying the imputation performance, this work focused on the classification to confirm if the conclusions were the same. It was expected that more complex datasets would obtain a higher classification performance, in this case, a higher F1-score. An approach proposed by Shin et al. [11], that preprocesses the data

considering simultaneously missing values and class imbalance, was used in this stage. The algorithm was also modified to consider only certain types of minority samples to infer if different types of points have different impact on the classification. The results were expected and different from the preliminary ones. The classification performance has a high correlation to the IR and the minority class distribution, the higher the IR and the percentage of unsafe samples, the lower the performance will be. In general, the less complex a dataset is, the higher the classification performance will be.

A possible future direction is to study different datasets' characteristics, like overlapping metrics. A possible way to improve the results is to tune the parameters differently for each dataset taking into account its complexity. With the obtained results, it might be interesting to develop an algorithm that performs the classification considering the relation between distribution and feature-based problems.

This page is intentionally left blank.

References

- [1] Saqib Ejaz Awan, Mohammed Bennamoun, Ferdous Sohel, Frank Sanfilippo, and Girish Dwivedi. Imputation of missing data with class imbalance using conditional generative adversarial networks. *Neurocomputing*, 453:164–171, 2021.
- [2] Gregory Piatetsky-Shapiro. Knowledge discovery in real databases: A report on the ijcai-89 workshop. *AI Magazine*, 11(4):68, Dec. 1990.
- [3] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, Mar. 1996.
- [4] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. Data preprocessing for supervised learning. *International Journal of Computer and Information Engineering*, 1(12):4104 – 4109, 2007.
- [5] Swagatam Das, Shounak Datta, and Bidyut B. Chaudhuri. Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, 81:674–693, 2018.
- [6] J. R. Quinlan. Improved estimates for the accuracy of small disjuncts. *Mach. Learn.*, 6(1):93–98, jan 1991.
- [7] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explor. Newsl.*, 6(1):40–49, jun 2004.
- [8] Sebastian Jäger, Arndt Allhorn, and Felix Biekmann. A benchmark for data imputation methods. *Frontiers in Big Data*, 4:48, 2021.
- [9] Paulo Oliveira, Fátima Rodrigues, and Pedro Rangel Henriques. A formal definition of data quality problems. In *ICIQ*, 2005.
- [10] Cátia M. Salgado, Carlos Azevedo, Hugo Proença, and Susana M. Vieira. *Missing Data*, pages 143–162. Springer International Publishing, Cham, 2016.
- [11] Kyoham Shin, Jongmin Han, and Seokho Kang. Mi-mote: Multiple imputation-based minority oversampling technique for imbalanced and incomplete data classification. *Information Sciences*, 575:80–89, 2021.

- [12] Pedro J. García-Laencina, José-Luis Sancho-Gómez, and Aníbal R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2009.
- [13] Adriana Costa, Miriam Santos, Jastin Soares, and Pedro Henriques Abreu. *Missing Data Imputation via Denoising Autoencoders: The Untold Story: 17th International Symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, October 24-26, 2018, Proceedings*, pages 87–98. 01 2018.
- [14] Melissa Azur, Elizabeth Stuart, Constantine Frangakis, and Philip Leaf. Multiple imputation by chained equations: What is it and how does it work? *International journal of methods in psychiatric research*, 20:40–9, 03 2011.
- [15] José M. Jerez, Ignacio Molina, Pedro J. García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, 50(2):105–115, 2010.
- [16] Qian Wang, Weijia Cao, Jiawei Guo, Jiadong Ren, Yongqiang Cheng, and Darryl N. Davis. Dmp_mi: An effective diabetes mellitus classification algorithm on imbalanced data with missing values. *IEEE Access*, 7:102232–102238, 2019.
- [17] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011.
- [18] Fei Tang and Hemant Ishwaran. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6):363–377, 2017.
- [19] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- [20] Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henrigues Abreu, Helder Araujo, and Joao Santos. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. *IEEE Computational Intelligence Magazine*, 13(4):59–76, 2018.
- [21] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review, 2013.
- [22] Vicente García, Josep Sánchez, Mollineda R.A, Roberto Alejo, and José Sotoca. The class imbalance problem in pattern classification and learning. *II Congreso Español de Informática*, 01 2007.

-
- [23] Xiaochen Lai, Yidan Lu, Liyong Zhang, Yi Feng, and Genglin Zhang. Imbalanced-type incomplete data fuzzy modeling and missing value imputations. In *2021 The 5th International Conference on Machine Learning and Soft Computing*, ICMLSC'21, pages 33–37, New York, NY, USA, 2021. Association for Computing Machinery.
- [24] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.
- [25] Adnan Amin, Sajid Anwar, Awais Adnan, Muhammad Nawaz, Newton Howard, Junaid Qadir, Ahmad Hawalah, and Amir Hussain. Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access*, 4:7940–7957, 2016.
- [26] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, jun 2002.
- [27] Alberto Fernández, Salvador García, Francisco Herrera, and Nitesh V. Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Int. Res.*, 61(1):863–905, jan 2018.
- [28] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, jun 2004.
- [29] Robert C. Holte, Liane E. Acker, and Bruce W. Porter. Concept learning and the problem of small disjuncts. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'89, page 813–818, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [30] Duke Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explorations*, 6:40–49, 06 2004.
- [31] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Learning with class skews and small disjuncts. In *Brazilian Symposium on Artificial Intelligence*, pages 296–306. Springer, 2004.
- [32] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explor. Newsl.*, 6(1):40–49, jun 2004.

- [33] J. Ross Quinlan. Improved estimates for the accuracy of small disjuncts. *Machine Learning*, 6(1):93–98, 1991.
- [34] Richard Taylor. Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1):35–39, 1990.
- [35] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768, 2018.
- [36] Maria-Carolina Monard. Learning with skewed class distributions. 04 2003.
- [37] Nitesh V. Chawla. *Data Mining for Imbalanced Datasets: An Overview*, pages 853–867. Springer US, Boston, MA, 2005.
- [38] Ricardo Barandela, Josep Sánchez, Vicente García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36:849–851, 03 2003.
- [39] Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5), sep 2019.
- [40] Krystyna Napierala, Jerzy Stefanowski, and Szymon Wilk. Learning from imbalanced data in presence of noisy and borderline examples. pages 158–167, 06 2010.
- [41] Krystyna Napierala and Jerzy Stefanowski. Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46, 07 2015.
- [42] José A. Sáez, Bartosz Krawczyk, and Michał Woźniak. Analyzing the over-sampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178, 2016.
- [43] Tianyu Liu, Wenhui Fan, and Cheng Wu. A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset. *Artificial Intelligence in Medicine*, 101:101723, 2019.
- [44] Ezgi Can Ozan, Ekaterina Riabchenko, Serkan Kiranyaz, and Moncef Gabbouj. An optimized k-nn approach for classification on imbalanced datasets with missing data. *Lecture Notes in Computer Science Advances in Intelligent Data Analysis XV*, pages 387–392, 2016.

-
- [45] Arjun Puri and Manoj Kumar Gupta. Knowledge discovery from noisy imbalanced and incomplete binary class data. *Expert Systems with Applications*, 181:115179, 2021.
- [46] Shigang Liu, Jun Zhang, Yang Xiang, and Wanlei Zhou. Fuzzy-based information decomposition for incomplete and imbalanced data learning. *IEEE Transactions on Fuzzy Systems*, 25(6):1476–1490, 2017.
- [47] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [48] Kaggle Datasets. <https://www.kaggle.com/datasets>, 2018. Accessed: 2022.
- [49] Giuseppe Casalicchio, Jakob Bossek, Michel Lang, Dominik Kirchhoff, Pascal Kerschke, Benjamin Hofner, Heidi Seibold, Joaquin Vanschoren, and Bernd Bischl. OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, pages 1–15, 2017.
- [50] Szymon Wojciechowski and Szymon Wilk. Difficulty factors and preprocessing in imbalanced data sets: An experimental study on artificial data. *Foundations of Computing and Decision Sciences*, 42, 01 2017.
- [51] Heikki Junninen, Harri Niska, Kari Tuppurainen, Juhani Ruuskanen, and Mikko Kolehmainen. Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, 38(18):2895–2907, 2004.
- [52] Steven J. Hadeed, Mary Kay O’Rourke, Jefferey L. Burgess, Robin B. Harris, and Robert A. Canales. Imputation methods for addressing missing data in short-term monitoring of air pollutants. *Science of The Total Environment*, 730:139140, 2020.
- [53] Edesio Alcobaça, Felipe Siqueira, Adriano Rivolli, Luís P. F. Garcia, Jefferson T. Oliva, and André C. P. L. F. de Carvalho. Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 21(111):1–5, 2020.
- [54] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705, 2008.
- [55] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), 2020.

Appendices

This page is intentionally left blank.

Appendix A

Complexity Metrics Description

Metric	Group	Description
<i>f1</i>	Feature-based	Maximum Fisher's discriminant ratio.
<i>f1v</i>	Feature-based	Directional-vector maximum Fisher's discriminant ratio.
<i>f2</i>	Feature-based	Volume of the overlapping region.
<i>f3</i>	Feature-based	Compute feature maximum individual efficiency.
<i>f4</i>	Feature-based	Compute the collective feature efficiency.
<i>l1</i>	Linearity	Sum of error distance by linear programming.
<i>l2</i>	Linearity	Compute the OVO subsets error rate of linear classifier.
<i>l3</i>	Linearity	Non-Linearity of a linear classifier.
<i>lsc</i>	Neighbourhood	Local set average cardinality.
<i>n1</i>	Neighbourhood	Compute the fraction of borderline points.
<i>n3</i>	Neighbourhood	Error rate of the nearest neighbor classifier.
<i>n4</i>	Neighbourhood	Compute the non-linearity of the k-NN Classifier.
<i>density</i>	Network	Average density of the network.
<i>cls_coef</i>	Network	Clustering coefficient.

Table A.1: Complexity metrics description. Adapted from *pymfe* [53].

Appendix B

Datasets Description

Dataset	Size	IR	Dataset	Size	IR
<i>appendicitis</i>	106	4,048	<i>neuthyroid2</i>	215	5,143
<i>audit</i>	775	1,682	<i>page_blocks0</i>	5472	8,789
<i>banana</i>	5300	1,231	<i>page_blocks_1_3_vs_4</i>	472	15,86
<i>bands</i>	365	1,704	<i>pageblocks_1_2vs_3_4_5</i>	5473	22,69
<i>banknote-authentication</i>	1372	1,249	<i>pageblocks_1vs3_4_5</i>	5144	21,27
<i>banknote</i>	1372	1,249	<i>pageblocks_1vs4_5</i>	5116	24,2
<i>bc-cosimbra</i>	116	1,231	<i>parkinson</i>	195	3,062
<i>biomed</i>	194	1,896	<i>phoneme</i>	5404	2,407
<i>breast-adi</i>	106	3,818	<i>pima</i>	768	1,866
<i>breast-car</i>	106	4,048	<i>poker_9_vs_7</i>	244	29,5
<i>breast-fad</i>	106	6,067	<i>prnn_synth</i>	250	1
<i>breast-gla</i>	106	5,625	<i>real-estate</i>	414	1,07
<i>breast-mas</i>	106	4,889	<i>redwine-2c</i>	1599	1,149
<i>cleveland_0_vs_4</i>	173	12,31	<i>relax</i>	182	2,5
<i>ctg-nvssp</i>	2126	3,514	<i>ring</i>	7400	1,02
<i>ctg-pathologic</i>	2126	11,08	<i>segment0</i>	2308	6,015
<i>ctg10</i>	2126	9,792	<i>shuttle_6_vs_2_3</i>	230	22
<i>ctg2</i>	2126	2,672	<i>somerville</i>	143	1,167
<i>ctg5</i>	2126	28,53	<i>sonar</i>	208	1,144
<i>ctg6</i>	2126	5,404	<i>spambase</i>	4601	1,538
<i>ctg7</i>	2126	7,437	<i>spectf</i>	267	3,855
<i>dermatology-3vs1</i>	182	1,563	<i>sports</i>	1000	1,74
<i>dermatology1</i>	358	2,225	<i>steel-plates-faults</i>	1941	1,884
<i>dermatology2</i>	358	4,967	<i>thyroid_3_vs_2</i>	703	18
<i>dermatology3</i>	358	4,042	<i>toy</i>	1250	1
<i>dermatology4</i>	358	6,458	<i>transfusion</i>	748	3,202
<i>dermatology5</i>	358	6,458	<i>twonorm</i>	7400	1,002
<i>dermatology6</i>	358	16,9	<i>urban-asphalt</i>	675	10,44
<i>ecoli1</i>	336	3,364	<i>urban-building</i>	675	4,533
<i>ecoli2</i>	336	5,462	<i>urban-car</i>	675	17,75
<i>ecoli3</i>	336	8,6	<i>urban-concrete</i>	675	4,819
<i>ecoli4</i>	336	15,8	<i>urban-grass</i>	675	5,027
<i>ecoli_0_1_4_6_vs_5</i>	280	13	<i>urban-pool</i>	675	22,28
<i>ecoli_0_1_4_7_vs_2_3_5_6</i>	336	10,59	<i>urban-shadow</i>	675	10,07
<i>ecoli_0_1_4_7_vs_5_6</i>	332	12,28	<i>urban-soil</i>	675	18,85
<i>ecoli_0_1_vs_2_3_5</i>	244	9,167	<i>urban-tree</i>	675	5,368
<i>ecoli_0_1_vs_5</i>	240	11	<i>user-know-h</i>	403	2,951
<i>ecoli_0_2_3_4_vs_5</i>	202	9,1	<i>user-know-vl</i>	403	7,06
<i>ecoli_0_2_6_7_vs_3_5</i>	224	9,182	<i>user-know-ullvshm</i>	403	1,251
<i>ecoli_0_3_4_7_vs_5_6</i>	257	9,28	<i>vehicle0</i>	846	3,251
<i>ecoli_0_3_4_vs_5</i>	200	9	<i>vehicle1</i>	846	2,899
<i>ecoli_0_4_6_vs_5</i>	203	9,15	<i>vehicle3</i>	846	2,991
<i>ecoli_0_6_7_vs_3_5</i>	222	9,091	<i>vertebral-h</i>	310	4,167
<i>ecoli_0_6_7_vs_5</i>	220	10	<i>vertebral-n</i>	310	2,1
<i>electrical-stability</i>	10000	1,762	<i>vertebral-s</i>	310	1,067
<i>forest-d</i>	523	2,289	<i>vowel0</i>	988	9,978
<i>forest-h</i>	523	5,081	<i>waveform-v1-0vs2</i>	3353	1,024
<i>forest-o</i>	523	5,301	<i>waveform-v1-1vs0</i>	3304	1,006
<i>forest-s</i>	523	1,682	<i>waveform-v1-1vs2</i>	3343	1,03
<i>glass0</i>	214	2,057	<i>waveform-v2-1vs0</i>	3345	1,024
<i>glass1</i>	214	1,816	<i>waveform-v2-1vs2</i>	3308	1,001
<i>glass2</i>	214	11,59	<i>waveform-v2-2vs0</i>	3347	1,022
<i>glass4</i>	214	15,46	<i>wdbc</i>	569	1,684
<i>glass5</i>	214	22,78	<i>whitewine-2c</i>	4898	1,987
<i>glass_0_1_2_3_vs_4_5_6</i>	214	3,196	<i>wifi1</i>	2000	3
<i>glass_0_1_4_6_vs_2</i>	205	11,06	<i>wifi2</i>	2000	3
<i>glass_0_1_5_vs_2</i>	172	9,118	<i>wifi3</i>	2000	3
<i>glass_0_1_6_vs_2</i>	192	10,29	<i>wifi4</i>	2000	3
<i>glass_0_1_6_vs_5</i>	184	19,44	<i>wine-1vs2</i>	130	1,203
<i>haberman</i>	306	2,778	<i>wine-3vs1</i>	107	1,229
<i>har-user4</i>	7918	2,64	<i>wine-3vs2</i>	119	1,479
<i>hepato-phusald</i>	294	1,534	<i>winequality_red_4</i>	1599	29,17
<i>hepato-phuslc</i>	302	1,435	<i>wisconsin</i>	683	1,858
<i>hillvalley</i>	1212	1,02	<i>wdbc</i>	198	3,213
<i>ionosphere</i>	351	1,786	<i>yeast1</i>	1484	2,459
<i>iris0</i>	150	2	<i>yeast3</i>	1484	8,104
<i>letter_u</i>	20000	23,6	<i>yeast4</i>	1484	28,1
<i>letter_z</i>	20000	26,25	<i>yeast_0_2_5_6_vs_3_7_8_9</i>	1004	9,141
<i>leukemia</i>	100	1,041	<i>yeast_0_2_5_7_9_vs_3_6_8</i>	1004	9,141
<i>liver-disorders</i>	345	1,379	<i>yeast_0_3_5_9_vs_7_8</i>	506	9,12
<i>lsvt-voice</i>	126	2	<i>yeast_0_5_6_7_9_vs_4</i>	528	9,353
<i>magic</i>	19020	1,844	<i>yeast_1_4_5_8_vs_7</i>	693	22,1
<i>new-thyroid-n-vs-hh</i>	215	2,308	<i>yeast_1_vs_7</i>	459	14,3
<i>neuthyroid-v1</i>	185	4,286	<i>yeast_2_vs_4</i>	514	9,078
<i>neuthyroid-v3</i>	180	5	<i>yeast_2_vs_8</i>	482	23,1

Table B.1: Properties of the datasets used in the experiments.

Appendix C

Artificial Datasets Results

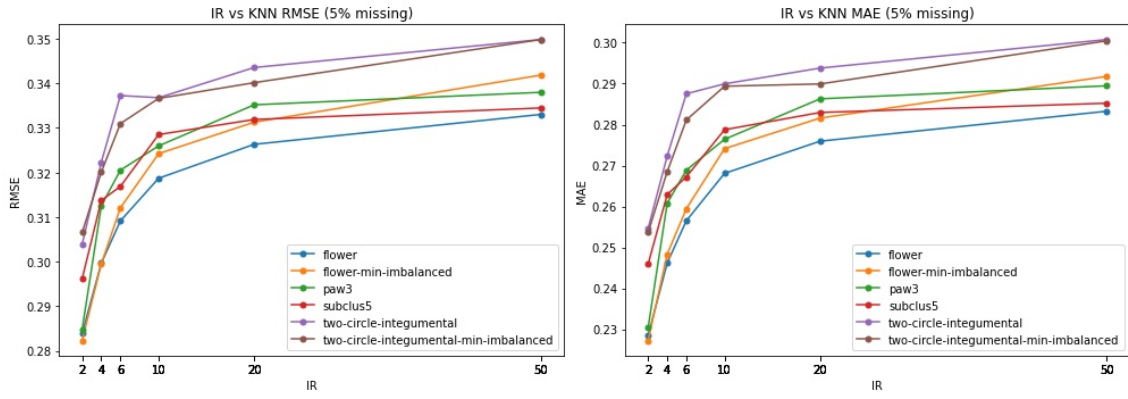


Figure C.1: Imputation error of the artificial datasets using k-NN with 5% of missing values.

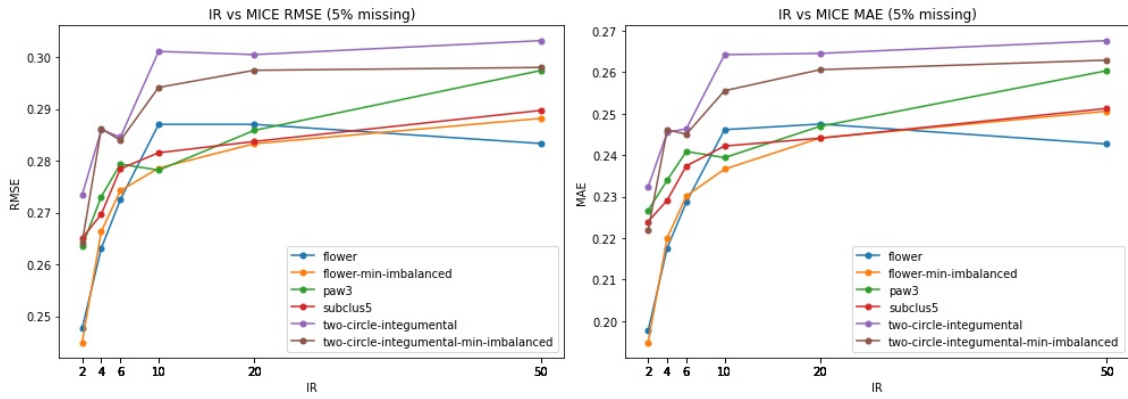


Figure C.2: Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 5% of missing values.

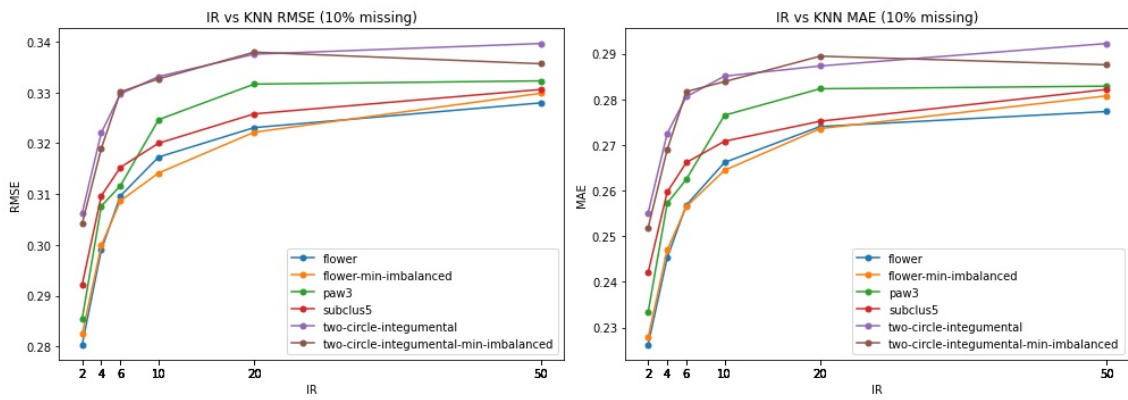


Figure C.3: Imputation error of the artificial datasets using k-NN with 10% of missing values.

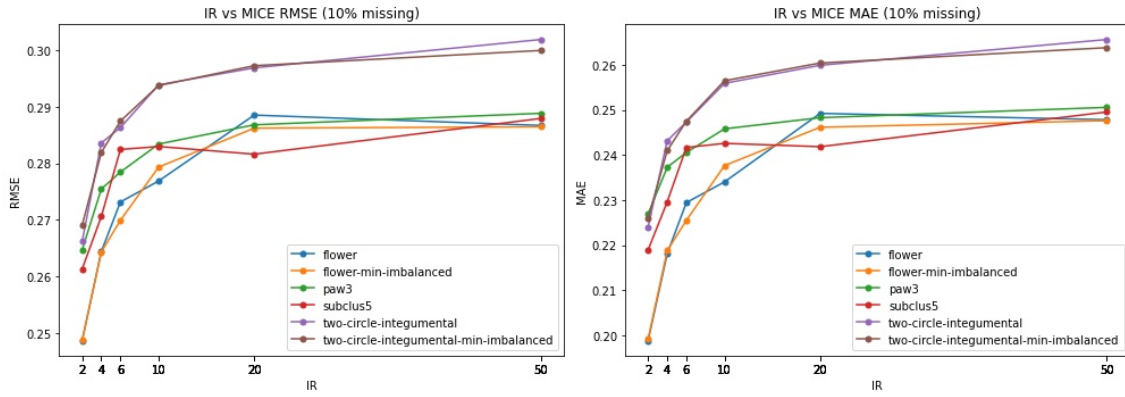


Figure C.4: Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 10% of missing values.

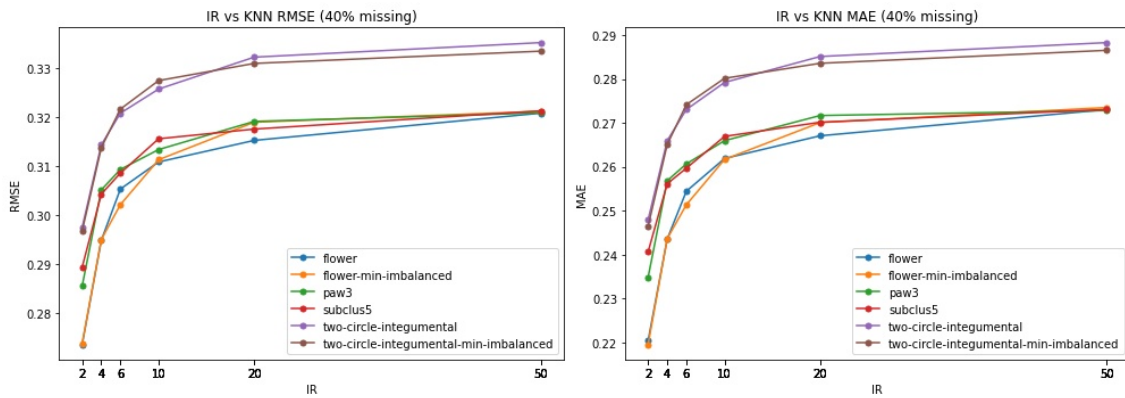


Figure C.5: Imputation error of the artificial datasets using k-NN with 40% of missing values.

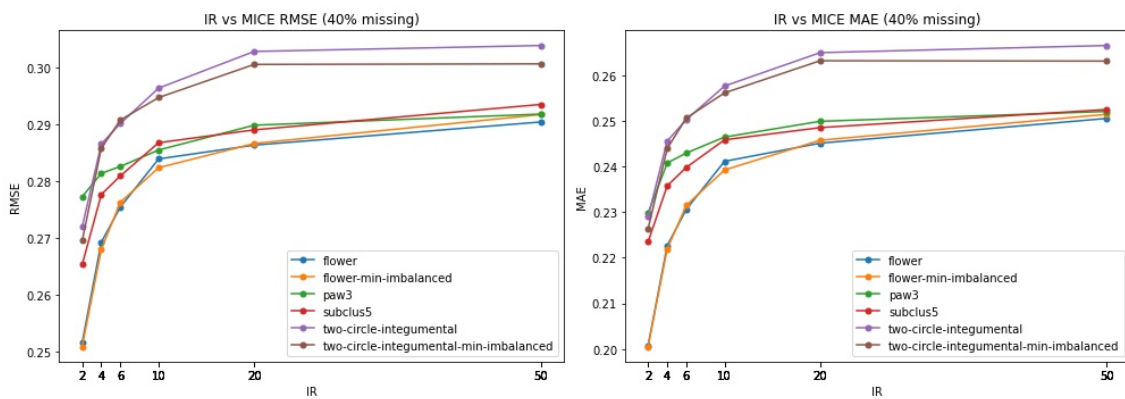


Figure C.6: Imputation error of the artificial datasets using MICE (with Linear Regression as estimator) with 40% of missing values.

Appendix D

Correlation for the Imputation Results

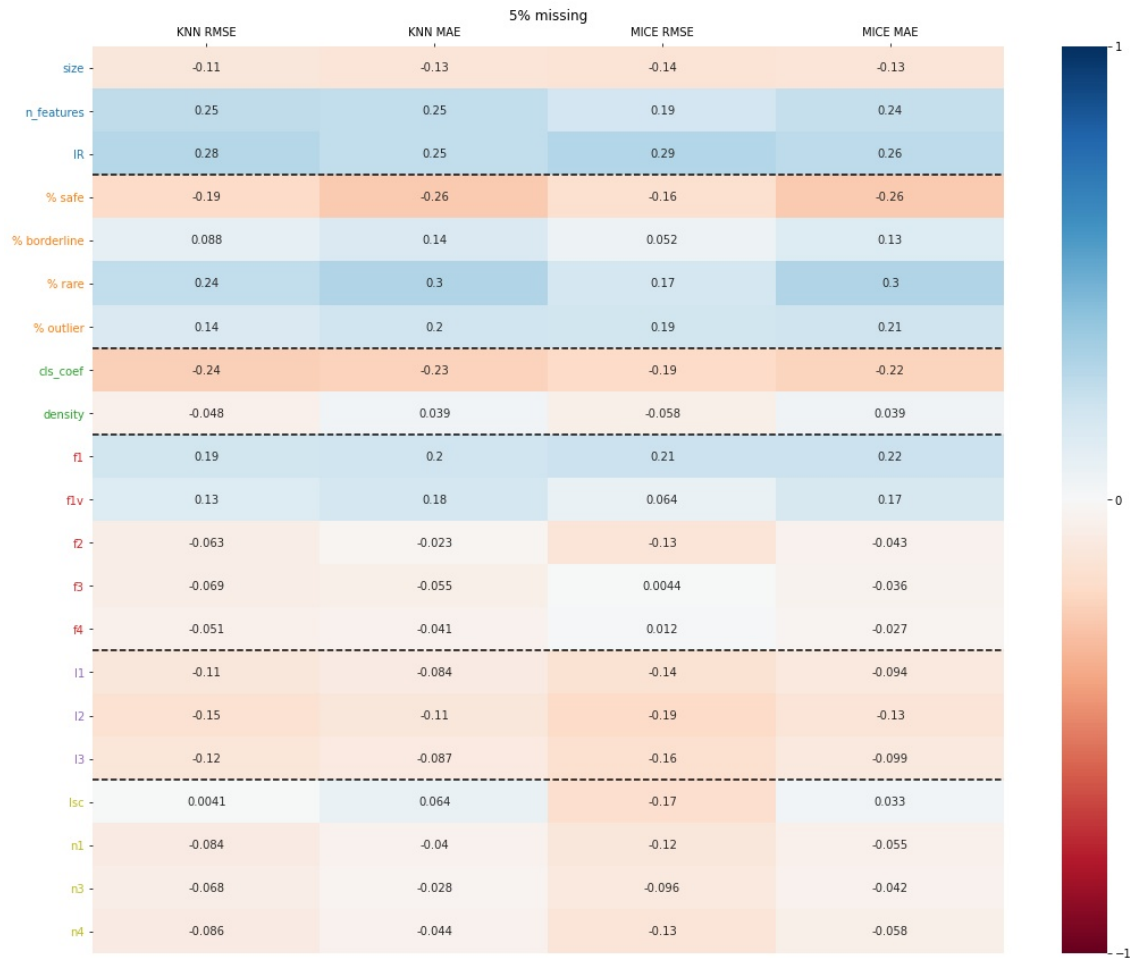


Figure D.1: Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 5% missing.

Correlation for the Imputation Results

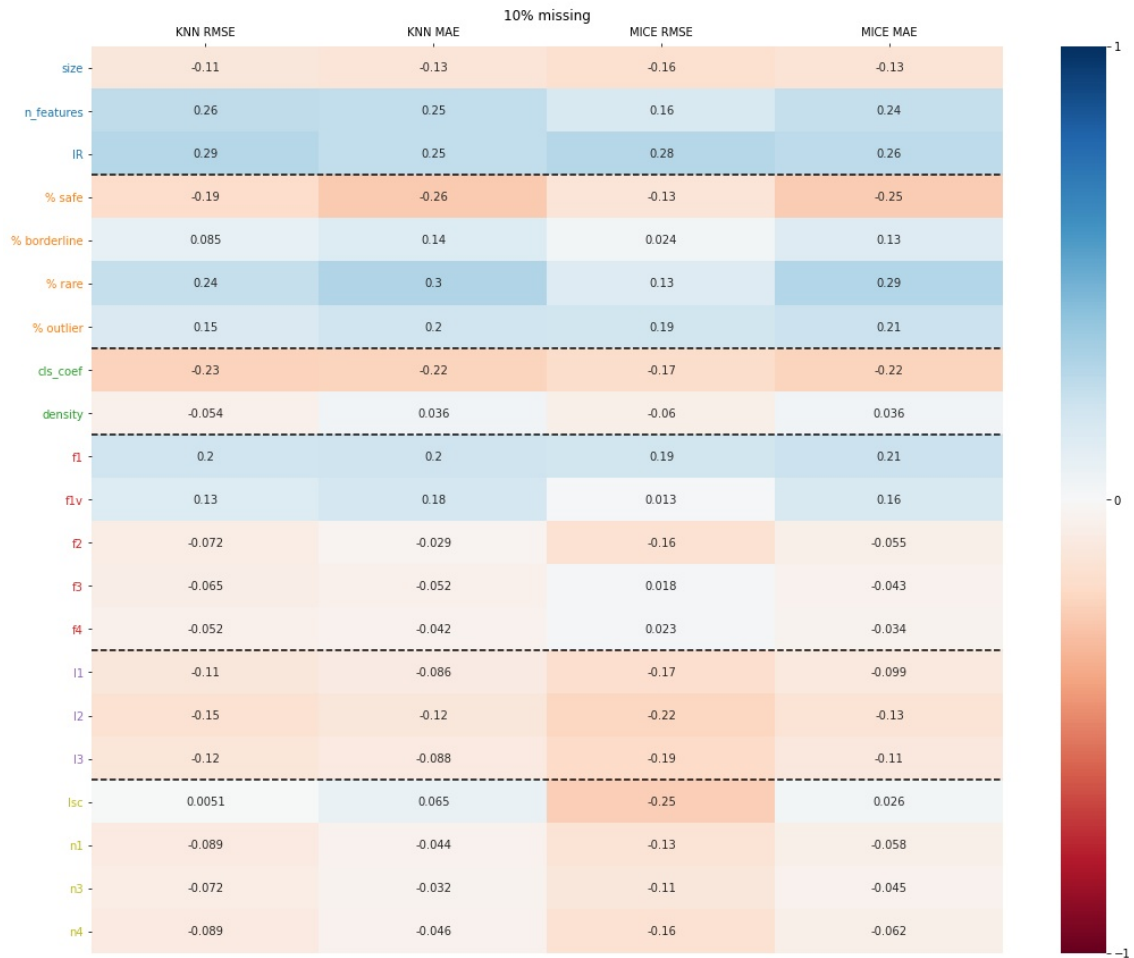


Figure D.2: Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 10% missing.

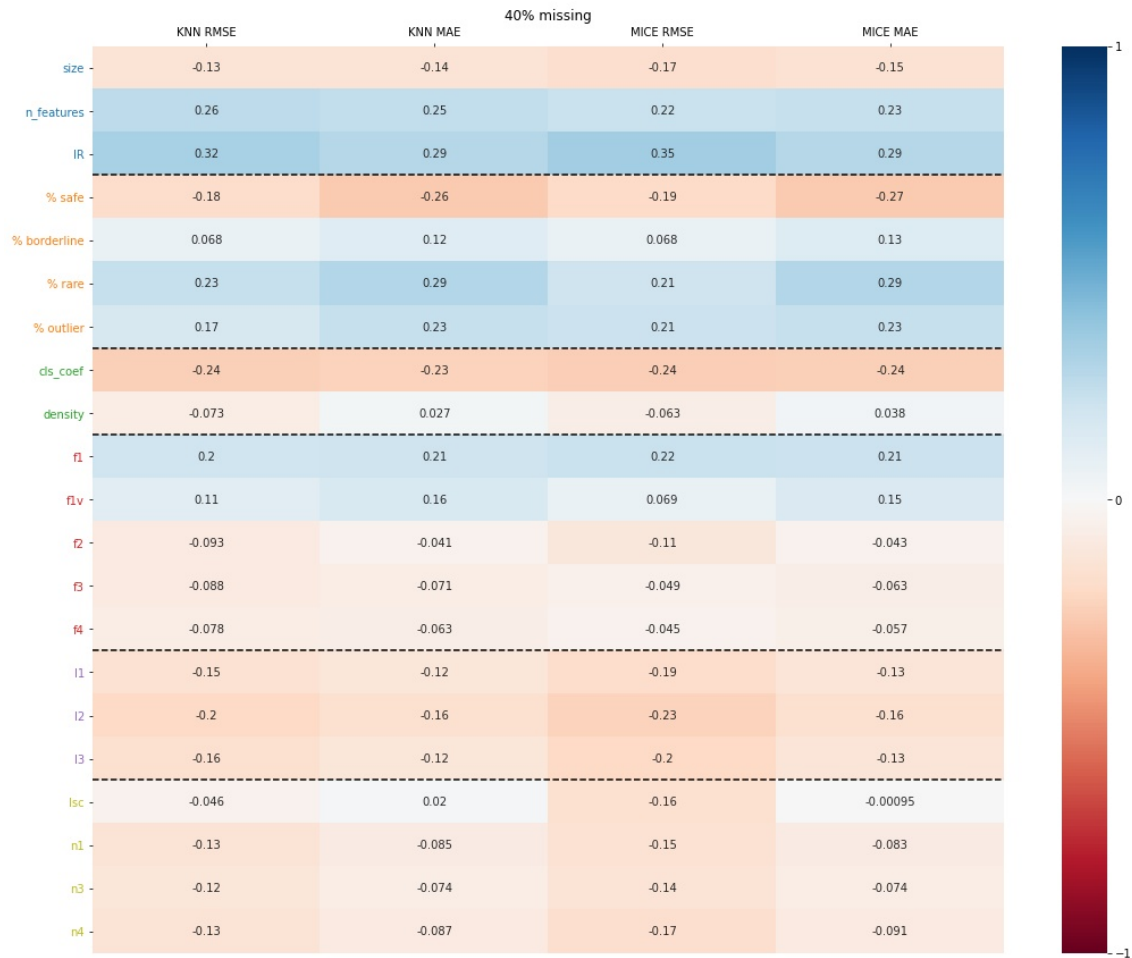


Figure D.3: Pearson Correlation Coefficient between the imputation error and all datasets's characteristics with 40% missing.

Appendix E

Correlation for the Classification Results

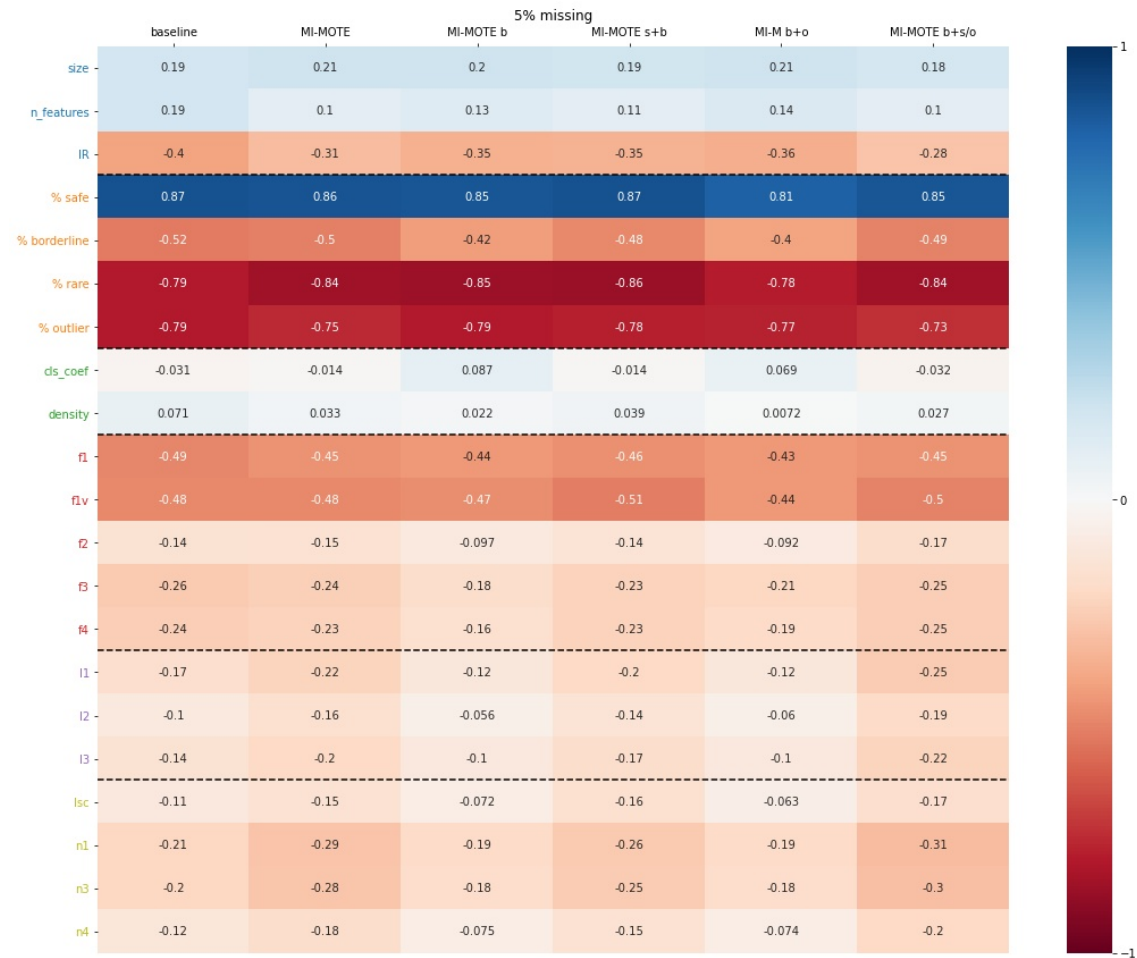


Figure E.1: Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 5% missing.

Correlation for the Classification Results



Figure E.2: Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 10% missing.



Figure E.3: Pearson Correlation Coefficient between the F1-score and all datasets's characteristics with 40% missing.