Cost-Sensitive Learning for Class Imbalance Data

A dissertation submitted to
Jawaharlal Nehru University, New Delhi
in partial fulfilment for award of the degree of

Master of Technology

in

Computer Science & Technology

by

Sai Teja Tangudu
(21/10/MT/016)

under the supervision of

Professor Rajeev Kumar

School of Computer and Systems Sciences
Jawaharlal Nehru University New Delhi

June 2023

Dedicated to my research idol Prof. Rajeev Kumar. . . .

School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi 110 067, India

Certificate

This is to certify that the dissertation entitled **"Cost-Sensitive Learning for Class Imbalance Data"** submitted by **Sai Teja Tangudu** (Enrollment No. 21/10/MT/016) to **Jawaharlal Nehru University New Delhi** towards partial fulfillment of requirements for the award of the degree of Master of Technology in Computer Science and Technology is a record of bonafide work carried out by him under my supervision and guidance during Academic Year, 2022 - 23.

(Dean)
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi

(Supervisor)
School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi

Acknowledgments

First and foremost, I extend my deepest appreciation to my supervisor, Prof. Rajeev Kumar. His constant support, exceptional guidance, and availability for discussions have been invaluable throughout my thesis. Prof. Rajeev Kumar's profound knowledge, expertise, and unwavering belief in my abilities have had a significant impact on shaping the outcome of my research. He has provided me with the freedom to explore various directions and encouraged me to delve into uncharted territories, which has positively influenced the depth and quality of my work. I am truly grateful for his exceptional mentorship and the confidence he has instilled in me to pursue a research career.

I would like to extend my sincere appreciation to Akhilesh Rawat, Bhupendera Kumar, and all my lab mates for their technical expertise and insightful interactions, which have greatly enriched my understanding of the subject matter. Their collective contributions have been invaluable to my research journey. I also want to acknowledge the significant contributions of Law Kumar and Biraja Mishra. Their critical insights and constructive feedback have expanded my intellectual horizons and shaped the direction of my research. Their valuable input has been instrumental in enhancing the quality of my work.

I am truly fortunate to have had such exceptional individuals as part of my academic journey, and I express my deepest gratitude for their invaluable support.

School of Computer and Systems Sciences
Jawaharlal Nehru University
New Delhi 110 067, India

Declaration

I certify that

1. The work contained in this report has been done by me under the guidance of my supervisor.

2. The work has not been submitted to any other Institute for any degree or diploma.

3. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

4. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Sai Teja Tangudu

Date: (20/10/MT/016)

Abstract

Class imbalance occurs if the spread of the samples in the training data is primarily skewed towards a single class. The performance of most classifiers degrades in case of class imbalance. This work reviews the factors characterizing the imbalanced data sets degrading a classifier's performance. A review of the adoption of cost-sensitive machine learning algorithms for class imbalance follows this. These algorithms focus on inducing different cost values into the objective functions of standard machine learning algorithms for weighing different misclassification errors differently.

In this dissertation, we analyze two verticals. First, the performance analysis of Cost-Sensitive Learning (CSL) algorithms over varying degrees of imbalance using various levels of class weights. We test the hypothesis: varying cost values influence the performance of Context-Sensitive Logistic Regression (CSLR) while dealing with a higher degree of class imbalance. But this has little or no impact on well/almost balanced data, with the help of the error bars obtained from our experiments. Second, Empirical evaluation and comparison of CSL algorithms with their non-cost sensitive counterparts in handling various challenges of class imbalance using a set of pre-defined use cases.

The work presented in this dissertation empirically validates the effectiveness of CSL algorithms in handling the intrinsic characteristics of the imbalanced data: small sample size of the minority class, data fragmentation, and skewed distribution at varying degrees of imbalance and how they compared to the conventional algorithms. Our analysis shows that cost-sensitive machine learning algorithms significantly outperformed the others in most use cases.

Contents

Title **i**

Acknowledgments **vii**

Abstract **xi**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Introduction 1
1.2 Basics, Terminology, & Background 4
1.3 Motivation 6
1.4 Issues & Challenges 7
1.5 Objectives 8
1.6 Organization of the Dissertation 9

2 Class Imbalance : An Overview **11**
2.1 Class Imbalance 12
2.1.1 Skewness 13
2.1.2 Sample Size 13
2.1.3 Class Separability 14
2.1.4 Data Fragmentation 15
2.2 Methods for Imbalance 15
2.2.1 Data Level Methods 15
2.2.2 Algorithemic Level Methods 17
2.3 Cost Sensitive Learning 18
2.3.1 Cost Matrix Design 18
2.3.2 Modification of Learners 19
2.4 Metrics for Class Imbalance 20

3 Cost Sensitive Algorithms : Empirical Evaluation **25**
3.1 Methodology Formulation for Evaluation 26
3.1.1 Construction of Use Cases for Empirical Analysis 26
3.1.2 Work Flow 28
3.2 Empirical Results and Analysis 29
3.2.1 Datasets and Preprocessing 29
3.2.2 Experimental Setup and Parameter Setting 30

3.2.3 Performance Metrics . . . 31
3.2.4 Use Case Analysis . . . 31
3.2.4.1 Use Case I: Medical *vs.* Non-Medical Datasets . . . 31
3.2.4.2 Use Case II: Small *vs.* Large Datasets . . . 33
3.2.4.3 Use Case III: Tree-based Learning Models *vs.* Non-tree-based 35
3.2.4.4 Use Case IV: Low *vs.* High Imbalance-Ratio Datasets . . . 37
3.2.5 Discussion . . . 39
3.3 Conclusion . . . 40

4 Evaluation of Cost Matrices for Cost-Sensitive Learners 41
4.1 Cost Matrices and their Utility in CSLs . . . 42
4.2 Related Work . . . 43
4.2.1 Designing of Cost Matrix . . . 43
4.2.2 Modification of Learners . . . 44
4.3 Methodology for Assessment of Cost Matrices . . . 45
4.3.1 Imbalancing Groups . . . 45
4.3.2 Sampling for Imbalancing . . . 45
4.3.3 Selection of Weights . . . 45
4.3.4 Cost-Sensitive Logistic Regression (CSLR) as the Reference Model . . . 45
4.3.5 Measures for Performance Assessment . . . 46
4.4 Results . . . 46
4.4.1 Experimental Setup . . . 46
4.4.2 Residual Errors . . . 47
4.4.3 Performance Measures . . . 47
4.4.4 Discussion . . . 48
4.5 Conclusion . . . 49

5 ADASYN-based Cost Matrix for Cost Sensitive Classifiers 51
5.1 Dataset Complexity Measures for Imbalanced Data . . . 52
5.2 Review of Existing Complexity Measures . . . 52
5.3 Proposed ADASYN-based Sample Weighting Method . . . 56
5.4 Methodology for Performance Assessment . . . 57
5.4.1 Cost Matrix Construction . . . 57
5.4.2 Model Training . . . 58
5.4.3 Measures for Performance Assessment . . . 58
5.5 Results and Discussion . . . 59
5.5.1 Data sets and Pre-processing . . . 59
5.5.2 Performance Assessment . . . 59
5.5.3 Discussion . . . 61
5.6 Conclusion . . . 62

6 Conclusion and Future Work 63
6.1 Work Summary . . . 63
6.2 Research Contributions . . . 64
6.3 Directions for Future Work . . . 65

List of Publications 69

References 71

List of Figures

2.1 Structural Organisation of the Literature Overview 12

3.1 PR score Distribution Medical vs Non-Medical 32
3.2 PR rank Distribution Medical vs Non-Medical 33
3.3 PR score Distribution : Sample Size 34
3.4 PR rank Distribution: Sample Size 35
3.5 PR score Distribution Tree vs Non-Tree 36
3.6 PR rank Distribution Tree vs Non-Tree 37
3.7 PR score Distribution: IR 38
3.8 PR rank Distribution: IR 38

4.1 Error distribution for four datsets 47
4.2 Kappa Score Distribution for the datasets. 48

5.1 Performance Overview 60

List of Tables

3.1 Medical Datasets 30
3.2 Non-Medical Datasets 30

4.1 Kappa score on the Vehicle dataset 47

1

Introduction

1.1 Introduction

Most classification tasks in real-world applications, e.g., the detection of abnormalities in medical data [1], fraudulent credit card transactions [2], risk prediction in domains like cyber security [3], spam and non-spam classification [4], detection of corners in image processing and computer vision [5], etc. involve highly class-imbalanced data. The drastic depletion in the performance of the classifiers while dealing with them is a major concern for the Machine Learning (ML) research community. The methods devised to address class imbalance are of paramount importance. However, the lack of systematic analysis of these methods creates uncertainty regarding their usefulness and practicality.

There are two main approaches to addressing class imbalance: data-level and algorithmic

level. The data-level approach aims to address the issue by redistributing the underlying data reducing the skewness present. Skewness in the data is often cited as a significant factor contributing to the degradation of classifier performance in the literature [6]. As it is believed to be a key factor that affects performance. Some of these methods include approaches like partial resampling [7], SMOTE [8], and counterfactual data augmentation [9], among others. These methods aim to balance the sample sizes of the classes and improve the classification accuracy for imbalanced datasets.

Every solution to class imbalance falls under either of the two approaches, namely, data-level or algorithmic level. *data-level* methods focus on redistributing the underlying data as One of the key reasons attributed to the degradation of the classifier's performance in the literature is skewness in the data [6]. Most researchers suggested ways to resample the data reducing the skewness present as it is believed to be the key factor affecting the performance and thereby balancing sample sizes of the classes present [7], [8], [9], etc. But skewness of the data is not the sole reason for the degradation of the performance of conventional classifiers. The degradation can also be attributed to other intrinsic data characteristics like the small sample size of the minority class, high domain complexity, low-class separability, and data fragmentation [10], [11], [12]. Some data-level approaches like AdaSYN[13] tried to address these issues by varying the way they redistribute the samples, i.e., sampling based on the hardness of learning a sample.

In addition, *algorithmic-level* methods focus on altering the existing learner and eliminating its bias towards the majority class is developed to handle the cases of imbalance. These methods focus on the algorithm rather than changing the underlying distribution of the data [14]. Cost-sensitive learning (CSL) is the most popular approach applied at this level. These methods are often proven to be more effective than re-sampling methods. Yet, most works on the CSL algorithms focused on showcasing their effectiveness in handling the skewness aspect of the class imbalance problem [14]. Hardly any studies showed how effective CSL algorithms are in handling the aspects of the data beyond skewness. In this work, we examined the effectiveness of CSL algorithms in addressing other factors related to imbalanced data. Specifically, we analyzed their performance in handling challenges such as the small sample size of the minority

class, the separability between classes, and the imbalance ratio (i.e., skewness), and OTHERS (SPECIFY), etc.

In addition to the above, it is observed that the literature often lacks comprehensive discussions on the general behaviour of CSL algorithms; therefore, their utilization for addressing class imbalance is relatively limited. This creates a significant research gap that needs to be addressed in order to comprehend the utility of CSL algorithms in effectively tackling imbalance problems. In this work, in order to gain insights into the general behaviour of CSL algorithms. We specifically examined their impact on the modified algorithm type (tree-based and non-tree-based) and the data domain being studied (medical and non-medical). By investigating these factors, we aimed to understand how CSL algorithms influence different types of algorithms and different domains, thereby providing a broader understanding of their applicability and effectiveness.

Considering the above-listed gaps in the literature, we identify the following properties to arrive at effective and appropriate solutions for imbalanced classification. It is necessary to mention that a competent solution demands knowledge of at least the following three properties:

- Its ability to address at least one of the factors characterizing the imbalance alongside skewness,
- Practitioner's prior understanding of the usability and applicability of the solution in various scenarios, including underlying data or/and model, and
- Its ability to handle cases of imbalance better than conventional algorithms

Furthermore, we propose a novel ADASYN (Adaptive Synthetic Sampling) based complexity measure and extensively evaluate it to showcase its superior performance compared to existing complexity measures in the cost matrix design. First, we investigate using data set complexity measures tailored to assess the complexity of imbalanced datasets in the cost matrix design. Secondly, we analyse the impact of these cost matrices on the performance of the Cost-Sensitive Logistic Regression(CSLR) classifier over five datasets and show the effectiveness of the proposed measure of cost matrix design.

Simply put, in this study, we aim to address the problem of class imbalance by investigating various factors contributing to class imbalance and skewness. We then analyze the performance of cost-sensitive machine-learning algorithms to determine if they effectively handle these conditions. Our approach involves conducting an experimental study and systematically analyzing the algorithms across predefined scenarios. Furthermore, we explore the effectiveness of Data Set Complexity Measures in designing cost matrices. Through empirical analysis, we demonstrate the efficacy of our proposed Learning Difficulty based measure for designing cost matrices. Overall, our study encompasses understanding class imbalance, evaluating cost-sensitive algorithms, and investigating the role of Data Set Complexity Measures in cost matrix design. By comprehensively examining these aspects, we aim to contribute valuable insights into addressing class imbalance challenges and improving the performance of cost-sensitive learning algorithms.

1.2 Basics, Terminology, & Background

Degree of Imbalance Every dataset has an inherent imbalance ratio (IR) representing the skewness of the dataset in terms of classes present. The IR is defined as the ratio of the sample size of the majority class to that of the minority class, i.e.,

$$Imbalance\ Ratio\ (IR) = \frac{Number\,of\,Majority\,Samples}{Number\,of\,Minority\,Samples}$$

We refer to the Imbalance ratio of each variant generated from the dataset as the degree of imbalance. Our work considers various resampled dataset variants with different degrees of imbalance for analysis. We analyse the performance of the algorithms involving four degrees of imbalance, namely, extreme imbalance (5:95), moderate imbalance (15:85, 30:70), and zero-degree imbalance (50:50). In the case of extreme imbalance, a variant consists of 5% of the minority class samples and 95% samples of the majority class and vice versa. As methods used for the sampling cannot always generate the exact required number of samples, the degrees of imbalance mentioned are approximate instead being exact.

Inverse Class Distribution Ratio Cost-sensitive learning (CSL) focuses on inducing the cost values into the algorithm's learning process to penalize misclassification errors differently. In this work, we accomplish this by setting the class weight parameter in the learner's API to a ratio called Inverse Class Distribution Ratio (ICDR), i.e., the weight of the minority class will be the number of samples in the majority class and vice versa. This ratio is calculated for each variant and then provided to the algorithm regarding class weight. The optimal class weight for inducing cost sensitivity into an algorithm is an open research question. In this work, we restrict ourselves to the inverse class distribution ratio for inducing cost sensitivity as it is flexible with the distribution of classes.

Tomek Links Complexity Measure The Tomek links Complexity Measure is a metric used to evaluate the level of class overlap in a dataset [15]. It is based on Tomek links, which are pairs of a minority class instance and a majority class instance that are nearest neighbors to each other. This measure uses the Tomek links count and provides an indication of the degree of overlap between the minority and majority classes. A higher TLCM value suggests more class overlap, while a lower value indicates greater class separation.

Precision-Recall Rank We use the area under the Precision-Recall (PR) curve metric to measure the efficiency of the classifiers used in the analysis. This is similar to the area under the ROC curve, except that the area calculated is under the PR curve rather than the ROC curve. PR curves are superior to ROC curves in analysing cases of imbalance [16]. But, as we aggregate a set of classifiers ranging from simple logistic regression to a complex XGBoost at varied levels of imbalance, a simple distribution of scores may not be enough to represent the persisting trends. Therefore, we use the distribution of the PR rank to rank the classifiers on their respective PR scores and illustrate the distribution of their ranks. Since we use a set of ten classifiers for our analysis, every scenario is analyzed using BoxPlots, representing the distribution of the PR rank and the PR score.

Box Plots and Interpretation A BoxPlot is a graphical method representing the locality, variance, and skewness in the numerical data regarding quartile groups. In this work, we focus

on the Range, Inter Quartile Range, Mean, and Median of the distribution, derived using the obtained BoxPlots for our analysis. In a BoxPlot, the minimum and maximum values are represented using the edges of whiskers, and the difference between them represents the distribution range. Similarly, we can derive Inter Quartile Range as the difference between the medians of the first quartile and the third quartile. A rectangular box in a BoxPlot represents it. Outliers, sample points that do not fit into the distribution pattern, are typically represented with a diamond symbol outside the whiskers of a BoxPlot. We represent the Mean and Median of the distribution using a black circle and a horizontal line across the box representing the middle 50% of values, respectively. We take inspiration from the work of Williamson et al. [17] for interpreting and analyzing the obtained BoxPlots.

1.3 Motivation

Class imbalance, where one class significantly outweighs the others, can lead to biased and inaccurate models that favour the majority class. This imbalance hinders the effective identification and classification of minority instances, often of greater interest and importance. Cost-sensitive learning (CSL) has emerged as a promising approach to address the imbalanced class distribution by assigning different misclassification costs to different classes. Our study aims to investigate the effectiveness of CSL algorithms in effectively handling imbalanced datasets across different degrees and scenarios. By evaluating the performance of CSL algorithms against conventional algorithms, we seek to determine whether CSL consistently outperforms traditional approaches in mitigating the challenges of class imbalance.

Additionally, we intend to explore how the behaviour of cost-sensitive algorithms varies with different degrees of imbalanced data. Understanding these variations can provide insights into the adaptability and robustness of CSL algorithms in different imbalance scenarios. This knowledge is crucial for selecting appropriate algorithmic techniques and parameter settings to achieve optimal performance in real-world imbalanced datasets. Another aspect we aim to address is the impact of selecting class weights on the performance of cost-sensitive algorithms. Class weights are vital in determining the importance assigned to each class during

training. Investigating the effect of different weight configurations on algorithmic performance will shed light on the sensitivity of cost-sensitive algorithms to class weight selection and assist in identifying the most suitable weight settings for different imbalanced datasets.

Overall, our study's motivation lies in enhancing the understanding of class imbalance issues, exploring the effectiveness of cost-sensitive learning algorithms, assessing their performance across varying degrees of imbalance, and evaluating the influence of class weight selection. By addressing these issues, we aim to contribute to the development of robust and reliable solutions for effectively handling imbalanced datasets and improving classification performance in real-world applications.

1.4 Issues & Challenges

This study on class imbalance and cost-sensitive learning faces several issues and challenges that need to be considered. Some of the key challenges include:

1. Availability of suitable datasets: Obtaining high-quality and diverse datasets with imbalanced class distributions can be challenging. It is essential to ensure that the selected datasets accurately represent real-world scenarios and cover a wide range of imbalance degrees and class distributions.

2. Selection of appropriate evaluation metrics: Choosing the right evaluation metrics is crucial for assessing the performance of cost-sensitive learning algorithms. Imbalanced datasets require metrics that consider the majority and minority classes, such as precision, recall, F1-score, and area under the ROC curve (AUC). However, selecting the most appropriate metrics for a particular study can be subjective and vary based on the application domain.

3. Determining optimal class weight configurations: Finding the optimal class weight configurations can be complex. Different weight settings can lead to varying algorithmic performance. Determining the most suitable class weights that balance minority class importance and overall accuracy is a challenging task that requires careful experimentation and analysis.

4. Generalization of findings: The effectiveness of cost-sensitive learning algorithms and the impact of class imbalance can vary across different domains and datasets. Ensuring the generalizability of the study findings and drawing meaningful conclusions that hold true for a wide range of applications is a significant challenge.

5. Algorithmic complexity and scalability: Some cost-sensitive learning algorithms can be computationally intensive and may face scalability issues when dealing with large-scale datasets. Balancing the need for accurate classification with computational efficiency is a challenge that needs to be addressed.

We address these challenges through careful experimental design, robust statistical analysis, and domain-specific considerations. We acknowledge these issues and take steps to mitigate their potential impact to ensure the validity and reliability of the findings and conclusions of this study.

1.5 Objectives

The objectives of this study are:

1. To analyze the effectiveness of cost-sensitive learning (CSL) algorithms in handling intrinsic characteristics of imbalanced data.

2. To compare the performance of CSL algorithms with conventional algorithms in handling imbalanced datasets.

3. To evaluate the impact of inducing cost sensitivity on classifier performance across various types of classifiers and different data domains.

4. To examine the behavior of cost-sensitive algorithms with varying degrees of imbalanced data.

5. To understand the impact of class weight selection on algorithm performance in cost-sensitive learning.

6. To provide insights and guidance for algorithm selection and implementation in real-life applications involving imbalanced datasets.

1.6 Organization of the Dissertation

The rest of the Dissertation is organized as follows,

Chapter 2 provides an in-depth analysis of the class imbalance problem, including a survey of existing literature on solutions and an overview of cost-sensitive learning algorithms. It covers the definition of class imbalance, analysis of intrinsic characteristics of imbalanced data, evaluation of data-level and algorithmic-level methods, and the performance metrics for evaluation of imbalanced cases.

Chapter 3 focuses on the empirical analysis of cost-sensitive learning algorithms. We investigate the effectiveness of these algorithms in handling the intrinsic characteristics of imbalanced data through a set of predefined use cases. The performance of both cost-sensitive (CSL) and cost-insensitive (CISL) algorithms is evaluated across varying degrees of imbalance using fifteen datasets from diverse domains. This chapter presents a comprehensive performance evaluation and discusses the effectiveness of cost-sensitive algorithms in addressing the class imbalance.

Chapter 4 aims to understand the significance of cost matrices in cost-sensitive classifiers. We examine the impact of cost matrices on the performance of the reference model, Cost-Sensitive Logistic Regression (CSLR), across different degrees of imbalanced data. This chapter provides insights into how the selection of class weights influences the classifier's performance.

Chapter 5 introduces a novel complexity measure based on Adaptive Synthetic Sampling (ADASYN) and evaluates its effectiveness in a cost matrix design. We explore the utilization of data set complexity measures for assessing the complexity of imbalanced datasets and analyze the impact of these cost matrices on the performance of the Cost-Sensitive Logistic Regression (CSLR) classifier using five datasets. The proposed measure demonstrates its effectiveness in a cost matrix design.

Chapter 6 concludes the dissertation by summarizing the research contributions and discussing future directions for further exploration in handling class imbalance and best practices for cost-sensitive learning.

2

Class Imbalance : An Overview

This chapter overviews the class imbalance problem, the solutions proposed in the literature survey, and the cost-sensitive learning (CSL) algorithms. This includes defining class imbalance, analyzing intrinsic characteristics of the imbalanced data, defining data-level and algorithmic-level methods from the literature, assessing their effectiveness in addressing the class imbalance, and computing various performance metrics developed for assessing the cases of imbalance.

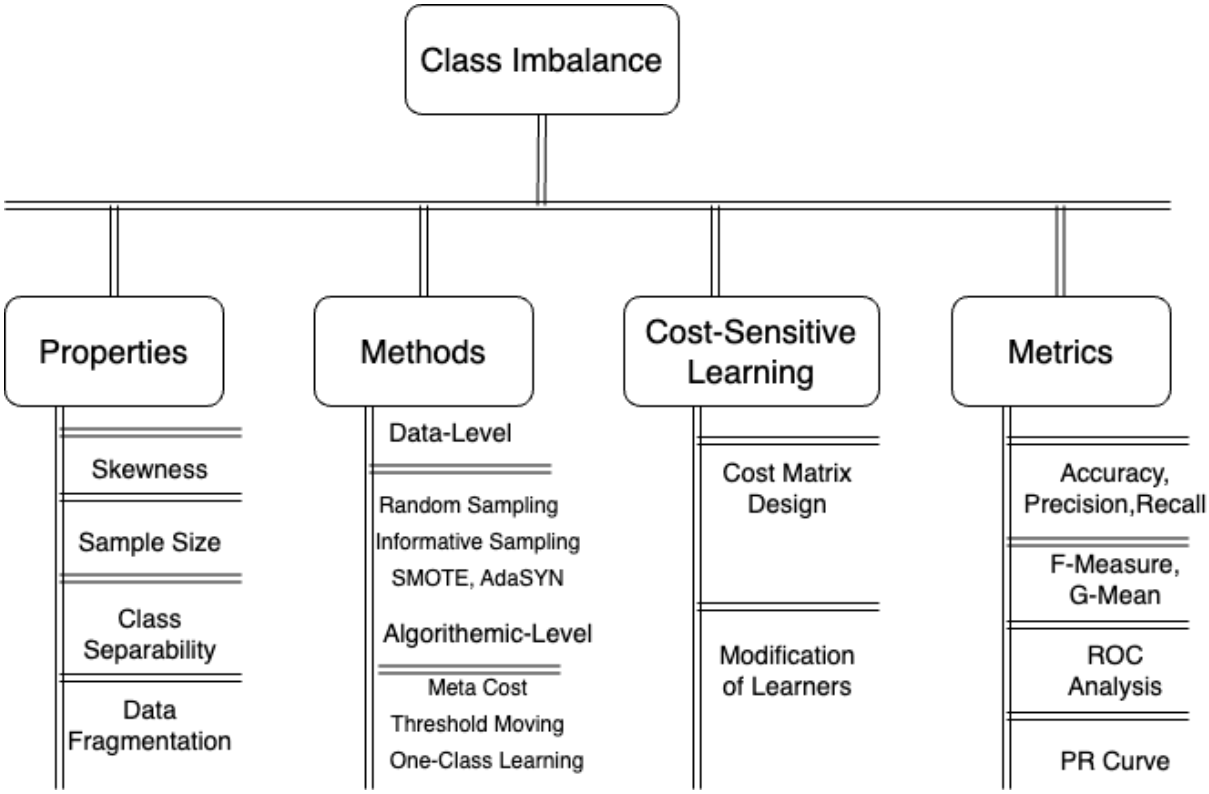


Figure 2.1: Structural Organisation of the Literature Overview

2.1 Class Imbalance

Class imbalance is a term that refers to a dataset having a skewed distribution of samples among its classes. The performance of most machine learning algorithms significantly drops in cases involving imbalance [14]. As the classifier is overwhelmed by the large sample size of the majority class, its performance is slanted towards the majority class in most cases. Therefore learning from a highly imbalanced dataset is said to be difficult [18], [19]. Most studies on the class imbalance problem attributed this significant drop in the classifier's performance to the skewness aspect of the data. However, Weiss [20] and Japkowicz [21] analysed the class imbalance problem and found that skewed distribution may not be the only factor affecting the performance of the classifier. They observed that it could also be attributed to some significant inherent characteristics of the dataset, including a small sample size of the minority class, class separability, and data fragmentation [10], [11], [12]. In this paper, we continue our research discussion to understand these intrinsic characteristics of the data along with the skewness aspect of the data.

2.1.1 Skewness

Skewness is a major characteristic and the root of the class imbalance problem [22]. Conventionally, the skewness of a dataset is characterised by the class imbalance ratio, which is defined as the ratio of the sample size of the majority class to that of the minority class or its inverse. Weiss and Provost, in their work on decision trees [23], analysed the impact of the skewness on the performance of the classifier and concluded that decision trees had given better performance on balanced class distributions than on datasets with imbalanced class distributions. Similarly, Japkowicz and Stephen [22] concluded in their study on the class imbalance problem, stating that with the increase in the degree of class imbalance, the complexity of the underlying concept of the data increases for the classifier. Singh and Kumar [24] empirically showed that the accuracy of conventional machine learning algorithms reduces with an increase in the degree of imbalance. However, this phenomenon cannot be generalized. According to Gottcke et al. [15], there is no strong correlation between the macro-precision or G-Mean scores of classifiers with the IR.

Henceforth, A classifier on a dataset with a very high imbalance ratio may give better results than a dataset with a relatively low imbalance ratio if other characteristics of the dataset, like class separability and sample size of the minority class, are in favor of the classifier [10].

2.1.2 Sample Size

In the case of classification, the learner's objective is to learn a best-fitting curve that better separates samples of the classes. This task requires the model to approximate the underlying functions of the classes correctly. According to Ali et al. [25], with the increase in the number of class samples, the amount of information available regarding the underlying function of the class increases. This, in turn, may help the model better approximate the best-fitting function of a class. This is true even in cases of imbalance. Japkowicz and Stephen [22] showed empirically that with the increase in the training dataset's size, the classifier's sensitivity towards the class imbalance decreases. Along with this study, Cui et al. [26] also showed the importance of having a large dataset for training. But in class imbalance problems, approximating the underlying function of the minority class is challenging. According to Japkowicz and Stephen [22]

on two datasets with the same imbalance ratio, a model may classify the minority class better on the dataset with the larger sample size of the minority class among the two as identifying the rare instances will be easier in that case. Experimental results show that if given a large enough minority class available for a model's learning, it performs well in the class even at a very high degree of imbalance. When this is limited, the approximation may be a case of under-fitting the minority class.

2.1.3 Class Separability

Class separability or class overlapping can be defined as the degree to which the class is separable from each other [27]. This is also an issue with the class imbalance problem. In most of the classification tasks, the underlying functions of the classes are not so discriminative. This means there will be one or the other subspace where the classes will overlap [10]. According to Sun et al. [10], this increases the complexity of the problem as more sophisticated rules are required to separate the classes. Prati et al. [11] studied the relation between class imbalance and class overlapping and suggested that degradation of the classifier's performance is not directly related to the degree of imbalance but rather to the degree of class overlapping. This work also empirically showed a strong correlation between class overlapping and class imbalance. Similar to sample size, class separability can also enhance or diminish the performance of a classifier irrespective of the degree of imbalance [25], [28]. Experimental results by Vuttipittayamongkol et al. [29] strongly support this argument. Tomek Link Complexity Measure (TLCM) [15], which measures the degree of overlap between the positive and negative classes, shows a stronger correlation with the performance of the classifiers than with the IR supporting the same.

A minority class with a highly discriminative pattern is learned better than a not-so-discriminative one. Similarly, highly overlapped classes may negatively affect the performance of a classifier, whereas not-so-highly overlapped classes may not trouble the classifier's performance. Experimental results also show that classifiers are not sensitive to the degree of imbalance in linearly separable. With the increase in the degree of overlapping, the classifier's sensitivity to the degree of the imbalance increases [11].

2.1.4 Data Fragmentation

Most machine learning algorithms decompose the classification task into problems of a very small magnitude that results in a partitioned instance space with very small partitions. This approach is called divide and conquer. The decision tree algorithm is a very good example of the same. But the major issue with this approach is that it can lead to Data fragmentation. Friedman et al. [30] pointed out that data fragmentation is one of the two structural and representational problems associated with decision tree learning. Fragmentation can also happen in cases where more features need to be tested during the learning. This is a challenge for the model in learning the underlying function, as each partition contains a very small amount of data. According to Weiss [20], this is a major concern even in the imbalance problem as the lack of samples of the minority class in a specific partition may force the learner to bias its performance toward the class making up the majority. Therefore most iterative methods that follow the divide-and-conquer approach will face difficulties in case of a higher degree of imbalance. This also boosts the usage of the non-iterative, non-divide, and conquer methods in cases involving imbalance.

2.2 Methods for Imbalance

2.2.1 Data Level Methods

These methods focus on redistributing the imbalanced data to make it easy for the classifier to learn better. Random undersampling of the majority class and Random oversampling of the minority class are the popular choices to balance the data. These sampling methods are simple yet inefficient in most cases [10]. The major issue with these Random sampling methods is the lack of information about the optimal data distribution. In most cases, the optimal distribution of the classes is unknown. Visa and Ralescu [31] experimentally showed cases where the balanced distribution (50:50 imbalance ratio) is proven to be non-optimal. Henceforth, balanced distribution being optimal is subjective. Another issue being random sampling may under-sample or over-sample more on a sub-class concept and less on another. A more efficient method is

to find the sub-class concepts and then sample them individually to balance the distribution. However, this may not be feasible in terms of the cost involved in the analysis.

Several authors have proposed another set of methods called Informative sampling. These methods involve selecting a subset of the sampling of the target class and then re-sampling these subsets of samples rather than the entire class. The issue with these methods is the selection criteria involved [10]. In the case of distance-based selection methods, The samples of the majority class that are far from the minority samples contribute more to the underlying function of the majority class. Similarly, the samples that are present close to the minority class contributes more to the decision boundary. The part that needs to be selected is, again, subjective.

Besides these, a more challenging problem with the re-sampling methods mentioned by Fernando and Tsokos [32] is that every method involves either oversampling or undersampling. According to Mienye and Sun [14], any oversampling method may unintentionally induce noise into the original data. In contrast, any undersampling method may involve the removal of the samples, which otherwise is critical for the classifier's learning.

Data augmentation methods like *SMOTE* and *AdaSYN* are also popular choices for handling imbalances at the data level. Chawla et al. [8] proposed SMOTE, an interpolation technique to systematically over-sample the minority class by augmenting samples instead of following the well-known oversampling with-replacement approach. AdaSYN [13] works on the principle that the amount of oversampling about various samples should be weighed upon the difficulty level of learning from them, i.e., more data will be augmented for a sample that is harder to learn from and vice versa. Along with these, Temraz and Keane [9] proposed a method adopted from the *eXplainable AI* for data augmentation. That generates synthetic samples by creating counterfactual instances for the minority class rather than interpolating.

Yasinnik [33] proposed a method to augment new minority class instances for an imbalanced dataset using a deep meta-learning approach. This method involves training a classifier to identify the highly uncertain regions. Samples are then augmented around these regions of uncertainty. Augmented samples are inducted into the learning process, and the classifier learns from them explicitly. The Augmented samples position is learned iteratively using a predefined

routine and subsequently included in the learning process of the classifier. The selection and applicability of sampling and augmentation methods are subjected to the imbalance case at hand and cannot be generalized; the experiments by Burez and An den Poel [34] confirm this in their study on handling imbalance in churn prediction.

2.2.2 Algorithemic Level Methods

Every machine learning algorithm works with a set of explicit or implicit assumptions regarding the data. These assumptions may hinder a classifier's performance in generalizing the imbalanced data. These assumptions are called the inductive bias of the classifier. Algorithmic-level methods focus on choosing an appropriate inductive bias to handle the imbalance [10]. According to Moniz and Monteiro [35], this selection cannot be generalized for all the algorithms as the underlying inductive biases differ.

For one algorithm, this can be done by selecting proper class weights. It may require a detailed understanding of the classifier's learning process and then accommodating the necessary modifications for another. For instance, decision trees can be altered by adjusting the probabilistic estimate at the leaf nodes or by developing new techniques for pruning. While on the contrary, SVMs can use different misclassification costs for the different classes involved [10].

But a significant challenge in developing these algorithmic-level solutions is that these methods require a detailed understanding of the classifier's learning process and a comprehensive understanding of the cause, because of which it fails in cases of imbalance. Cost-sensitive algorithms are the most popular approaches applied at this level, discussed in Section 2.3. The following is the summary of the other popular methods: meta-cost, one-class learning, thresholding moving methods, and the associated works.

Domingos [36] proposed a meta-cost approach for solving the class imbalance. This meta-cost algorithm creates multiple replicates of the original training data and employs a classifier on each. Then, it estimates each class's probability based on the number of votes it received and relabels all the examples with the estimated class. As a final step, it reapplies the classifier on the relabeled training data. In case of any modification to the cost values, the algorithm must repeat only the final stage of the learning process instead of the remaining algorithms.

The threshold moving method shifts the output threshold close to majority class instances, making it difficult for the classifier to misclassify samples with higher costs. It employs cost values at the classification phase rather than the training phase. According to Maloof [37], this method is unpopular yet as effective as re-sampling techniques for solving imbalance. It inputs the real-valued outputs of a trained multilayer perceptron (MLP) and calculates the overall cost by multiplying the output with its corresponding misclassification cost. It returns the class with the highest overall cost as the output.

According to Sun et al. [10], One-class learning deals with cases involving imbalance by learning only from the minority class. These algorithms focus on approximating the decision boundary that surrounds the samples of the minority class, contrary to the other algorithms, which approximate a decision boundary partitioning the hypothesis space that separates the minority and the majority classes. These algorithms depend on the similarity between the samples and the target class and employ a threshold value for classification. This type of learning is shown to be effective in cases involving multi-modal domains. The pre-set threshold defines the boundary between the classes in these algorithms. Therefore, setting a proper threshold is crucial for their learning. Moreover, algorithms such as Naive Bayes and decision trees do not work unless the training data involve other classes.

2.3 Cost Sensitive Learning

Most cost-sensitive learning methods focus either on the cost matrix design or on the modification of the existing learner [38]. This section summarizes the related work in these verticals individually.

2.3.1 Cost Matrix Design

Cost-sensitive learning focuses on weighing different misclassification errors of a confusion matrix differently. The term "cost matrix" refers to a matrix with these weights. To compensate for the imbalance, false positives are weighted higher than true positives in most of these cost matrices. Since the entire work of a cost-sensitive algorithm depends on the weights of the cost

matrix, its design is of utmost importance. A domain expert typically validates the applicability of a cost matrix if one is accessible *a priori*. According to Moepya et al. [39], actual misclassification costs of these cost matrices are difficult to find. To evaluate the performance of their proposed cost-sensitive strategy in identifying financial fraud from transaction data, they used pre-determined cost matrices with eight levels of cost weights.

In this paper, we set the cost matrix values using a more prevalent method known as *inverse* class distribution ratio. This ratio was utilised by Mienye and Sun [14] to analyze cost-sensitive learning methods for imbalanced data from the medical domain. Krawczyk [40] criticised this technique of cost matrix design, claiming that weights must be chosen based on the difficulty of learning from a class rather than the classes' distribution. Since our work aspires to cover most aspects of the dataset through the predefined use cases, we used this distribution-based method in designing the cost matrix because of its simplicity and ease of application.

2.3.2 Modification of Learners

Objective functions used by most machine learning algorithms weigh misclassification errors on different classes equally. Cost items induced in cost-sensitive learning algorithms modify this inherent characteristic by mutating the weightage of these classification errors either according to the view of a domain expert or based on the distribution of the classes in the data. The motivation for these modified objective functions would be to reduce the overall cost rather than the training error during the learning.

Mienye and Sun, in their work on cost-sensitive learning for the classification of the imbalanced medical data [14], resolved on inducing cost matrices possessing varied misclassification costs into the classifiers. They considered the objective functions of the classifiers, namely logistic regression, SVM, decision tree, random forest, and XGBoost classifiers, and induced inverse distribution ratio into them. They showed the superiority of these algorithms over other methods for solving the imbalance. Sun et al. [10] developed three variants of the AdaBoost objective function by inducing cost items into its weight update formula for classifying the imbalanced data. They refer to these variants as AdaC1, AdaC2, and AdaC3. AdaC1 induces cost values outside its exponential function, whereas AdaC2 induces the same inside the ex-

ponential function. AdaC3 combines both approaches inducing cost values in and out of the exponential function. AdaC2 is shown to perform better in imbalance cases among the three empirically. Incorporating different classification metrics into the objective functions is another popular method applied at the algorithm level. Cao et al. [41] mutated the objective function of the SVM classifier using G-Mean and AUC, inducing cost sensitivity. Intrinsic parameters, feature subsets, and cost parameters are optimised simultaneously in this work to enhance performance.

2.4 Metrics for Class Imbalance

According to Sun et al. [10], choosing a sufficiently good performance metric can help enhance the efficiency of the machine learning algorithm in improving the search process and in the analysis of the result. Many of the metrics can be used for both. Most classification metrics, such as accuracy, precision, and recall, are proven inadequate for cases of imbalance. This section summarizes the analysis of the performance metrics for class imbalance from the literature.

The simplest and most intuitive metric for the evaluation of classifiers is accuracy. It is the most used performance metric for classification in the literature. But it cannot handle the cases of imbalance in many ways. A study on evaluation metrics for imbalance by Japkowicz [42] argues that accuracy weighs the majority class more than the minority class, reducing the classifier's efficiency in identifying the rare class instances. Gu et al. [43] also mentioned a shortcoming of accuracy: it cannot differentiate the types of errors it makes. This study also supports the use of precision/recall in imbalance cases over accuracy since they are more representative. It also pointed out the inability of precision/recall in separating the performance over different classes.

According to Provost and Fawcett [44], metrics such as accuracy assumes balanced class priors and equal error costs despite being unsure about the precise conditions of the environment in which a model is going to be deployed, makes them unsuitable for imbalance cases. Japkowicz stated this argument could be generalized even to metrics other than accuracy [42].

Since most objective functions focus on reducing the training error that inherently adopts accuracy, we require more sophisticated metrics for handling imbalance.

Kubat et al. [45] argued the use case of G-Mean for imbalance cases, as they consider the properties of the metric to be independent of the class distributions and is capable of impacting the cost of misclassifying an instance dynamically as noteworthy. According to them, the G-Mean dynamically varies the cost of misclassifying a positive example following the increase/decrease in the frequency of the misclassified positive samples. Gu et al. [43] also supported the usage of G-Mean for cases of imbalance as it simultaneously maximises the accuracy in both classes with a very good trade-off. We could not find any study contradicting the usage of G-Mean for cases of imbalance. According to Japkowicz [42], Other metrics like F-Measure that allow the user to adjust the weights of the precision and recall components offer more robustness than that of G-Mean.

ROC curves and the area under the ROC curves (AUC) are used for evaluating the classifier in most recent studies on imbalance. These are preferred over most other metrics as they emphasise equally on both classes without bias. According to Provost and Fawcett [44], ROC assesses the performance of the classifiers without the assumptions of balanced class priors and equal error costs. Therefore, they proposed ROC analysis as a more suitable metric for imbalance. Japkowicz [42] attributed two significant reasons for the superiority of ROC analysis in cases of imbalance: its ability to decompose the performance of each of the classes into two distinct measures rather than being a single multi-class focus metric and its inclusivity of the diverse cost ratio scenarios in a domain whose characteristics are unknown precisely beforehand.

Gu et al. [43] also studied the shortcomings of ROC analysis. According to them, ROC is superior to the other metrics in separating the classifier's performance in both classes. Yet, it can mask the inability of weak classifiers in cases of imbalance. Davis and Goadrich [46] argued that the ROC curve's optimism bias regarding a classifier's performance could suffer severely in cases of imbalance. Webb and Ting [47] contradicted the hypothesis of ROC analysis being stable and invariable to the change in environment by presenting how true positive and false positive rates vary with changing distributions. But Fawcett and Flach [48] argued

that this statement might only apply to one of the two classes present and may not always hold. Japkowicz [42] suggested a thorough understanding of both these works on ROC analysis [47], [48] before applying the ROC analysis in a changing class distribution scenario.

Precision-Recall (PR) curves, a visually strong representative space than ROC, is also proven helpful in cases of imbalance. As the name suggests, PR curves plot the precision of a classifier about the change in the degree of recall [42]. Similar to ROC plots, it explores the trade-off between the correctly classified positive samples to that of the misclassified negative samples. According to Wasikowski and Chen [49], the PR curve is a non-parametric statistic performance metric used to assess classifiers. Davis and Goadrich [46], in their study on the relationship between PR and ROC curves, systematically analysed some major aspects of similarity between them. First, They argued that both PR-Curve and ROC-Curve on a dataset contain the same points. This implies that any curve dominating the ROC space must dominate others in the PR space first. Second, this study demonstrated a sub-space of the PR curve called the achievable PR Curve, an equivalent of the convex hull of a ROC curve that helps find the optimal threshold point. These points add to the argument that PR-Curves have analogous benefits to ROC curves for performance assessment of classifiers [50].

Bradley et al. [51] recommended the usage of PR curves instead of ROC curves while dealing with environments that are not precisely known and do allow variability in class distributions. They attributed this to the fact that the major motivation in these environments would be to observe the variability in the classifier's performance. They argued this is done more accurately by PR curves than ROC. Davis and Goadrich [46] argued that there is a big difference in the visual representation of ROC and PR curves. According to their study, the PR space visually exposes the difference in algorithm performance, which is not apparent in the ROC analysis. Therefore, supporting the applicability of PR curves in cases involving highly imbalanced data may mask the performance of the weak classifier. Hancock et al. [52] strongly supported this argument and experimentally showed the inefficiency of AUC in cases involving highly imbalanced big data.

A dominant classifier in the ROC space whose performance is close to an ideal classifier may have a very poor performance in the PR space showing a high scope for improvement.

This work also argues that in these highly imbalanced cases PR curve can be more informative about the classifier's performance than ROC. This argument is also supported by the work of He and Garcia [50].

Bradley et al. [51] proposed two metrics that summarize the PR curve under different scenarios. These metrics are equivalents of what AUC is to ROC. The first metric summarizes the PR curve under a single prior, and the second is defined under a whole range of priors. This study empirically demonstrated that the sensitivity of these metrics to the variability in the distribution is higher than that of the AUC, strengthening the case PR curves for imbalance. This study uses the area under a PR curve under a single prior to analyze the classifiers. Even though the PR curve's usage is restricted to the cases of binary classification and the expansion of this to multi-class is hardly discussed in the literature, most of the works switched from ROC to PR curves for the assessment of cases of imbalance, e.g., [53], [54], [55], [52] and [16].

3

Cost Sensitive Algorithms : Empirical Evaluation

Cost-sensitive algorithms are often proven to be more effective than re-sampling methods. Yet, as the general behaviour of the CSL algorithms is rarely discussed in the literature, their utilisation for solving the imbalance is limited. Most works on the CSL algorithms focused on showcasing their effectiveness in handling the skewness aspect of the class imbalance problem [14]. But skewness may not be the sole reason for the degradation of the performance of conventional classifiers. The degradation can be attributed to other intrinsic data characteristics like the small sample size of the minority class, high domain complexity, low-class separability, and data fragmentation. Hardly any studies showed how effective CSL algorithms are in handling these aspects of the data besides skewness. This creates a huge research gap that needs to

be filled to understand the utility of CSL algorithms for solving imbalance problems.

In this chapter, we empirically analyse and discuss the effectiveness of CSL algorithms in handling these intrinsic characters of the data using varied scenarios called use cases. These use cases are defined by focusing on one or more characteristics. We analyse the performance of both cost-sensitive (CSL) and insensitive (CISL) algorithms over these use cases at varying degrees of imbalance to enhance the detail. We have considered fifteen datasets from diverse domains possessing varied characteristics with different levels of imbalance ratio for enhancing the hypotheses' generalizability. We consider cost-sensitive variants of logistic regression, SVM, decision tree, random forest, and XGBoost classifiers and their cost-insensitive counterparts for the analysis. We present a comprehensive performance evaluation of the mentioned cost-sensitive algorithms over fifteen datasets at varying degrees of imbalance under the defined use cases and discuss their effectiveness in handling the class imbalance.

3.1 Methodology Formulation for Evaluation

Most research on CSL algorithms focused on the IR factor of the imbalanced data. In this work, we empirically analyse their behaviour in handling other factors. Accordingly, in this section, we first define the problem and present the construction of use cases for the analysis. We then systematically present the steps of the proposed workflow.

3.1.1 Construction of Use Cases for Empirical Analysis

In this section, we define use cases that focus on one or the other data-based scenario that we use to examine the enhanced performance of CSL over CISL algorithms empirically. Every use case has an inheriting hypothesis mapped to it that acts as a query that investigates specific characteristics of the cost-sensitive algorithms under study. The set of experiments performed remains constant throughout the use cases. But every use case takes some/all results to analyse the mapped hypothesis. This section details the use cases considered for the analysis. Every use case is analysed in two verticals, and we refer to these verticals as aspects.

Use Case I: Medical Data This use case focuses on understanding the efficiency of inducing cost sensitivity into the conventional machine learning algorithms in dealing with highly complex medical domain data and comparatively less complex data from other domains. We split our analysis into two aspects, medical and non-medical, based on the general view in the studied literature that medical data often composes highly complex patterns and the costs associated with unfavorable consequences of developing lesser efficient models for medical diagnosis are severe. Out of the fifteen datasets considered for the analysis, seven are from the medical domain, and the rest belong to other domains like finance, banking, remote sensing, etc., which are considered under the non-medical aspect.

Use Case II: Sample Size This use case focuses on analysing the effectiveness of CSL algorithms over data of varied sample sizes of the minority class. We split our analysis into two aspects regarding the number of samples of the minority class in the dataset, small and large. Based on the evidence from the literature in Subsection 2.1.2, the difficulty in learning from the minority samples decreases with the increase in the number of minority samples present in the dataset, despite the increase in the degree of imbalance. Datasets that consist of more than one thousand minority class samples are considered under the large datasets aspect, and the rest comes under the aspect of small datasets.

Use Case III: Learning Models This use case analyses the effectiveness of inducing cost sensitivity into machine learning models prone to data fragmentation, i.e., tree-based models, and those not, i.e., non-tree-based models. According to Weiss [20], most divide-and-conquer approaches may see a drop in performance in cases of a higher degree of imbalance. From the list of classifiers considered for our analysis, random forest, decision tree, and XGBoost algorithms follow these approaches and therefore come under the tree-based models' aspect. In contrast, logistic regression and SVM are part of the non-tree-based models' aspect.

Use Case IV: Imbalance Ratio This use case focuses on the effectiveness of CSL algorithms over data of various degrees of skewness, i.e., varied imbalance ratios (IR). Two aspects, namely a higher and a lower Imbalance Ratio, are considered for the analysis. Datasets with an IR

greater than ten are considered under the higher IR aspect, and the rest are considered under the aspect of lower IR. This use case is defined based on the hypothesis from the literature [22] that the performance of conventional classifiers declines while dealing with the datasets with sufficiently higher inherent skewness/imbalance ratio detailed in Section 2.1.1.

3.1.2 Work Flow

Generating Imbalance Variants Each of the fifteen datasets is resampled to create four variants of the original version. Fifteen percent of the total samples are preserved as testing data, and the rest is used for generating these variants. The extreme imbalance variant has 5:95, the zero-degree imbalance variant has 50:50, and the other variants come with 15:85 and 30:70 degrees of imbalance that characterize moderate imbalance. We initially calculate the required number of samples needed to be present in a variant depending on the degree of imbalance. Then apply AdaSYN and random undersampling for either oversampling or undersampling a class, respectively, to construct the variant.

Models and Cost-Sensitive Counterparts We consider standard classes of logistic regression, decision tree, random forest, SVM, and XGBoost from the sklearn module for analysis. We refer to them as Cost Insensitive algorithms (CISL). We set the class weight to ICDR during the instantiation to induce cost sensitivity into these algorithms. We refer to them as cost-sensitive learning algorithms (CSL). All other hyperparameters available in the APIs are set to default values to reduce the experiments' variability.

Results Aggregation and Rank Calculation We analyze the mentioned cost-sensitive (CSL) algorithms and their five CISL counterparts' performance over all the datasets at the mentioned degrees of imbalance. Each of the ten algorithms is implemented over all variants of each dataset individually. The performance of an algorithm over a specific variant of a dataset is analyzed over the test data generated for the dataset, and the associated PR score is calculated. Each algorithm is then ranked based on its PR score for a specific variant of a dataset, i.e., the algorithm with the highest PR score is given a rank of 1. Two algorithms with the same PR score will be given the same rank, skipping the consequent rank.

Interpretation of Results We generate a set of BoxPlots for every use case, comparing the performance of both CSL and CISL algorithms. Every plot comprises the distribution of PR scores or the ranks obtained by these two types of algorithms that aggregate results on either a set of models or datasets as per the requirement of a use case. Every plot showcases the distribution of scores obtained by CSL algorithms that differ from their CISL counterparts over a specific degree of imbalance. We present the mean of the distribution along with the median for better analysis and inferencing. The higher the skewness of the distribution of the scores away from the horizontal axis, the better the algorithm is in the case of PR score distribution plots. The lower, the better in the case of plots showing PR rank distribution.

3.2 Empirical Results and Analysis

3.2.1 Datasets and Preprocessing

This section detail datasets used for empirical analysis. We use fifteen datasets from various domains, including seven from the medical domain (Table 3.1) and eight from various other domains (Table 3.2), i.e., insurance, finance, remote sensing, etc. Most of the datasets are taken from UCI[1] repository, and others are taken from various sources: Insurance [56], Credit fraud [57], Mammography [58], Portoseguro [59], Phenome [60], Pima [61], ECG [62], and Ninapro [63]. Most of the datasets considered are binary. In the case of multi-class datasets, We considered the class with the highest number of samples as the majority and the class with the least number of samples as the minority, turning the problem into binary. We provide the respective dataset details in Table 3.1 and Table 3.2. Most datasets contain NULL values; we used KNNImputer from sklearn for the imputation. We used MinMaxScaler from sklearn for scaling the values in the dataset.

[1] https://archive-beta.ics.uci.edu/

Table 3.1: Medical Datasets

Dataset	MinoritySample	MajoritySample	IR
abalone	42	689	16.4
ninapro	3815	63314	16.6
cervical	55	803	14.6
haberman	81	225	2.8
pima	268	500	1.9
ckd	150	250	1.7
ecg	2081	2919	1.4

Table 3.2: Non-Medical Datasets

Dataset	MinoritySample	MajoritySample	IR
credit Fraud	492	284,315	577.9
mammography	260	10923	42
portoseguro	21694	573518	26
covtype	2747	35754	13
insurance	62601	319,553	5
vehicle	199	647	3.3
phenome	1586	3818	2.4
ionosphere	126	225	1.8

3.2.2 Experimental Setup and Parameter Setting

We use Python 3.6 and *jupyter* notebook for implementation purposes. We use the *imblearn* module for all re-sampling-related tasks. This module includes methods like AdaSYN and random undersampling, which we use for oversampling and undersampling. The instances of *XGBClassifier, DecisionTreeClassifier, LogisticRegression, RandomForestClassifier,* and *LinearSVC* from *sklearn* are used with default settings and hyperparameters for implementing non-CSL algorithms. The same classifiers are used with the class weight parameter set to ICDR for the analysis of CSL algorithms except in the case of XGBoost where we provided the class weights through the *scale_pos_weight* parameter. The rest of the parameters are set to default for experimental convenience. Since the standard sklearn implementation of the SVM is slow, We use *sklearnex*, an extension of the standard sklearn module developed by *Intel*. We use the *patch_sklearn* method on the *Google colab* environment to ensure smoother execution

of SVM.

3.2.3 Performance Metrics

PR curve is a visual representation of the trade-off between precision and recall of a classifier's performance. These curves plot precision as a function of change in the recall representing them on Y and X axis, respectively. We present our analysis using two metrics defined over the PR curve. First, the area under the PR curve that we refer to as the PR score. Second, a Rank is calculated based on the area under the PR curve with respect to the other classifiers on a specific dataset as discussed in Section 1.2 that we refer to as the PR rank. PR curves are typically hyperbolic, just like the ROC curves. Except that a dominant curve in the PR space resides close to the upper right-hand side of the PR curve, unlike the ROC curve where it resides close to the upper left-hand side of the curve [50].

3.2.4 Use Case Analysis

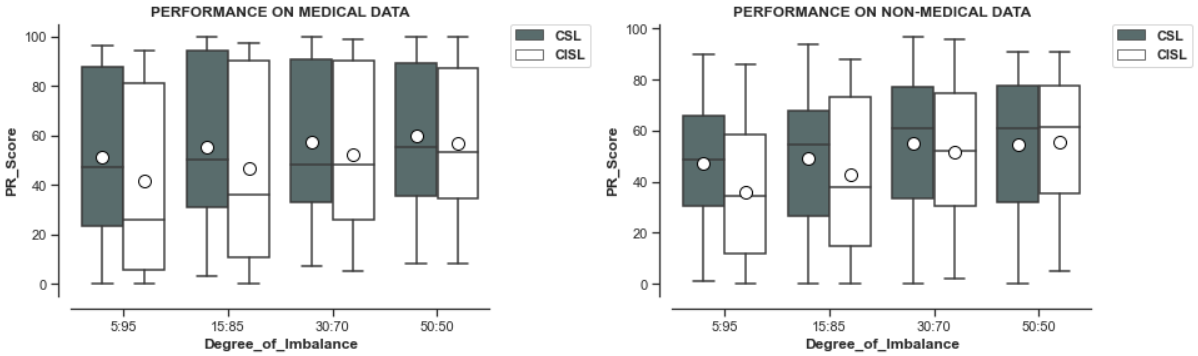
In this sub-section, we present the results concerning every use case considered. We present the results as BoxPlots representing the respective PR scores and rank distribution.

3.2.4.1 Use Case I: Medical *vs.* Non-Medical Datasets

CSL algorithms have outperformed their CISL counterparts on medical data in almost all performance measures. Firstly, They have higher median and mean over all the degrees of imbalance compared to their counterparts. The difference in their means and medians has increased with the increase in the degree of the imbalance, indicating a solid superiority of the CSL algorithms over them in handling cases of imbalance. Next, The range of the boxes alongside the interquartile range is higher in cases of CISL algorithms signifying higher variability and volatility in the performance compared to that of the CSL algorithms.

Besides, the mean and median of the CSL boxes also remained constant over all the degrees of imbalance, supporting the fact that CSL algorithms are comparatively more stable. They both have given similar results in the case of zero-degree imbalance variants signifying that inducing cost sensitivity into algorithms has no significant advantage over balanced data.

This can be attributed to the fact that when the distribution is equally distributed at the class level, the inverse class distribution ratio (ICDR) will almost be equal, i.e., 1:1, which means equal weightage to the classes. This is the default assumption of the base model. Inducing cost sensitivity at this degree of imbalance using inverse class distribution does not show any significant difference in performance.

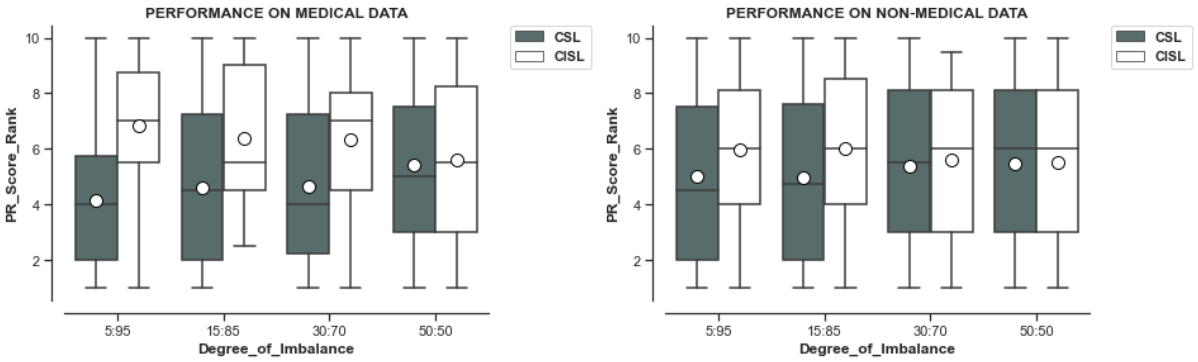


(a) On Medical Datasets (b) On Non-Medical Datasets

Figure 3.1: PR score Distribution Medical vs Non-Medical

CSL algorithms have significantly outperformed their counterparts while dealing with medical data, which the research community assumes to have inherently complex underlying patterns. So, This observation aids the utilisation of CSL algorithms for imbalanced tasks on the data from highly complex domains such as bio-medical over the conventional CISL algorithms. Similar performance trends regarding mean and median can also be observed over non-medical data. That is not the case regarding the range and the interquartile range. We can observe from Fig. 3.1b that the range of the CSL boxes is higher than those of their counterparts, showing higher variability in the overall distribution. Yet, the interquartile range of the same is comparatively lower, showing higher agreement among the middle 50% of the PR scores.

According to Rindskopf and Shiyko [64], the interquartile range does a better job than the range in describing the dispersion present in the data. This further supports the case that CSL algorithms are also more stable than their counterparts in dealing with non-medical data. The stable enhancement in the performance of CSL algorithms, irrespective of the change in assumed domain complexity, shows a strong signal for their utility in cases of imbalance, irrespective of this inherent characteristic of the data.



(a) On Medical Datasets (b) On Non-Medical Datasets

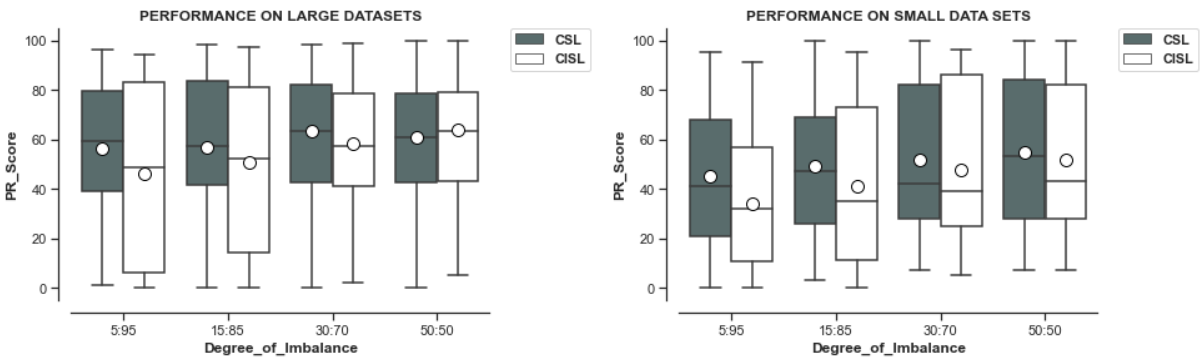
Figure 3.2: PR rank Distribution Medical vs Non-Medical

Fig. 3.2 represents PR rank distributions of the CSL algorithms for the same medical and non-medical data. These distributions show almost similar patterns and trends present in the PR score distributions, except that these plots show even stronger support for CSL algorithms in the aspect of medical data. Simply put, conventional algorithms could not handle data from highly complex domains like medicine, and inducing cost sensitivity is helping the cause. This is evident in both PR score distributions and PR rank distributions.

3.2.4.2 Use Case II: Small *vs.* Large Datasets

The PR scores plotted in Fig. 3.3 show that CISL classifiers are stable and perform well on large datasets, irrespective of the degree of imbalance. In the case of small datasets where the number of minority class samples available is less, there is a drastic decline in the performance of the CISL algorithms. This can be attributed to the evidence from the literature detailed in the use case description in Section 3.1.1.

Let us now understand the impact of inducing cost sensitivity into these algorithms with the help of PR scores from Fig. 3.3. In large datasets, CSL algorithms are better in terms of mean and median at all the degrees of imbalance except the zero-degree imbalance. Boxes of CSL algorithms have less interquartile range than their counterparts at higher degrees of imbalance, namely 5:95 and 15:85, even though they are not that better in terms of box range. These observations imply that when there are enough minority class samples available for learning, the imbalance cases seem manageable unless it gets extreme. As these patterns are not strongly depicted in the distributions, these observations cannot be generalized.

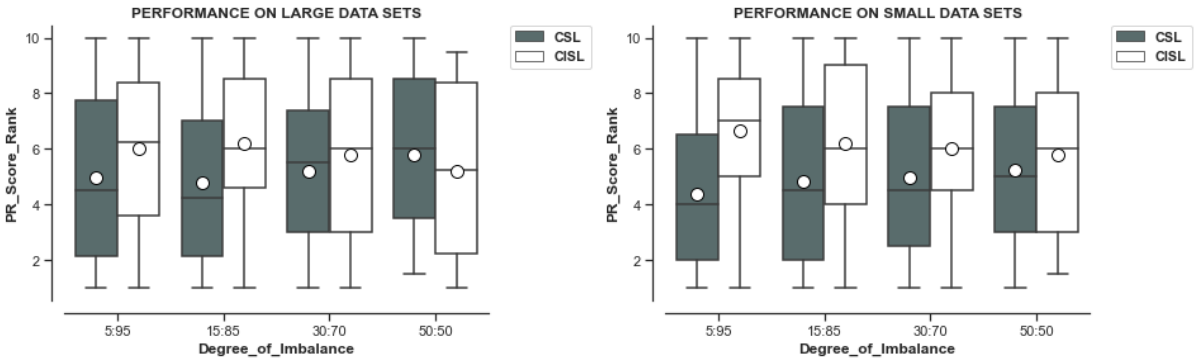


(a) On Large Datasets (b) On Small Datasets

Figure 3.3: PR score Distribution : Sample Size

In Fig. 3.4a, the dominance of CSL algorithms over their counterparts is strongly visible. They have better mean, median, and interquartile ranges skewed towards the horizontal axis and a relatively comparable box range at all the degrees of imbalance except for zero-degree imbalance. Inducing cost sensitivity helps the classifier improve its performance even with datasets with a large minority class sample size, except for balanced class distribution.

As mentioned, CISL algorithms' performance has significantly dropped in the case of small datasets depicting their dependency on the sample size of the minority class. This is true even in the case of CSL algorithms, except that the difference in the mean and median of the scores in the large and the small datasets aspect is high in the case of CISL algorithms compared to CSL algorithms. This implies that CSL algorithms handle the small size of the minority class aspect of the data better than the CISL algorithms. Therefore aiding their utility irrespective of the sample size of the minority class. But even the performance of CSL algorithms suffered in the case of small datasets. Their performance can be further improved by applying more robust methods for designing the cost matrix and inducing cost values. Similar results are reported by Japkowicz and Stephen in their systematic study on class imbalance [22].



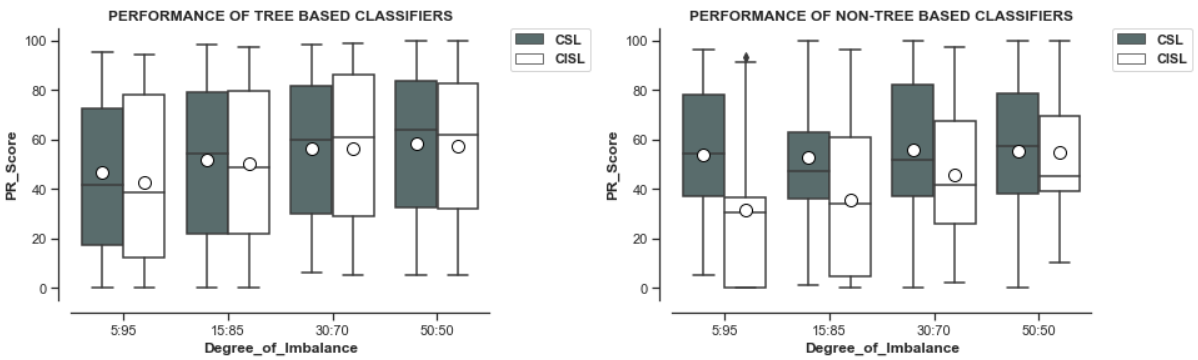
(a) On Large Datasets (b) On Small Dataets

Figure 3.4: PR rank Distribution: Sample Size

Another significant observation made is, when it comes to the degree of imbalance, there is an exception in the case of the zero-degree imbalance in the large datasets aspect, where CISL algorithms performed better. But this exception is negotiable since that is not a proper imbalance case as the ratio of samples is 50:50, implying that CSL algorithms may not suit well-balanced distributions. The same viewpoint is more strongly represented in PR rank plots 3.4 where there is a significant difference in the means and medians of the CSL and CISL boxes at the zero-degree imbalance in the large datasets aspect.

3.2.4.3 Use Case III: Tree-based Learning Models *vs.* Non-tree-based

Regarding tree-based learning models (Fig. 3.5), CSL algorithms performed better regarding mean and median. But in terms of range and interquartile range, their performance is compromised. At extreme imbalance, CISL models have an interquartile range skewed away from the X-axis, depicting better performance. Yet, they have a large interquartile range showing high variability and less agreement among the middle 50 percentile of the scores. At all other degrees of imbalance, there is a reasonable tradeoff between the cost-sensitive and insensitive model's performance. Similar tradeoffs are visible regarding PR ranks (Fig. 3.6). Therefore, no conclusive evidence can be found from the experiments to aid the positive impact of inducing cost sensitivity into tree-based models.



(a) On Tree-Based Models (b) On Non-Tree Based Models

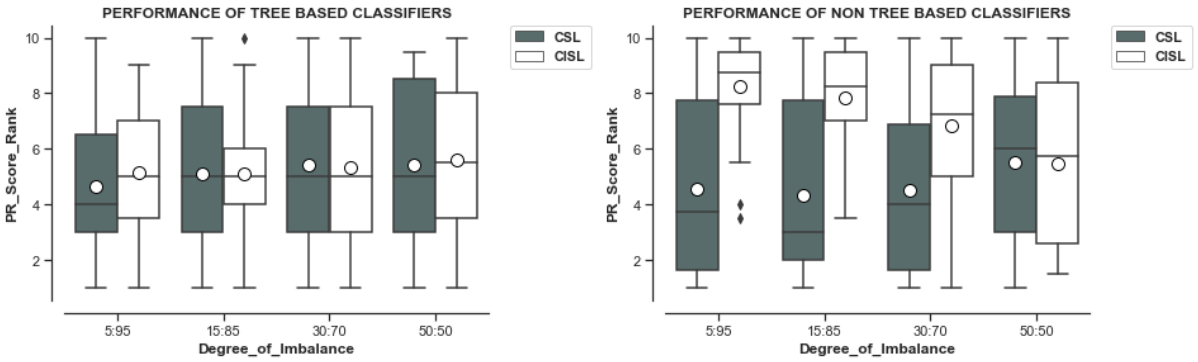
Figure 3.5: PR score Distribution Tree vs Non-Tree

Unlike tree-based models, substantial performance improvements are visible in non-tree-based models. From Figs. 3.5b and 3.6b, we can observe that non-tree-based models are sensitive to imbalances present in the data, and inducing cost sensitivity can enhance their performance by a distance. There is a significant improvement in the distribution of both PR scores and their ranks with the induction of cost sensitivity. Except in the zero-degree imbalance case, the performance declined after inducing cost sensitivity. This again validates that CSL algorithms may fail to perform well on balanced distributions.

As per the experimental results, we can observe that inducing cost sensitivity impacts different classifiers differently. Inducing cost sensitivity improved the performance of the classifiers from both aspects. But the performance improvement in non-tree-based classifiers is more strongly visible than in the case of tree-based classifiers. This proves that the impact of inducing cost sensitivity is more in non-tree-based algorithms than in tree-based algorithms. This is evident over all the degrees of imbalance in mean, median, range, and interquartile range.

Besides these observations, one should note that tree-based algorithms have performed better than non-tree-based algorithms before inducing cost sensitivity. This can be attributed to two possible reasons. Reason one, tree-based models may handle the cases of imbalance better than non-tree-based models. Reason two can be the case of no occurrence of data fragmentation during the classifier learning because tree-based models performed exceptionally well. Since reason one is out of the scope of this paper, it is important to focus on reason two. This implies that the frequency of data fragmentation is lower in real-world use cases. Suppose this argument about the non-occurrence of data fragmentation is true. In that case, these experi-

ments are not enough to comment on whether CSL algorithms can handle this aspect of the classifiers.



(a) On Tree-Based Models (b) On Non-Tree Based Models

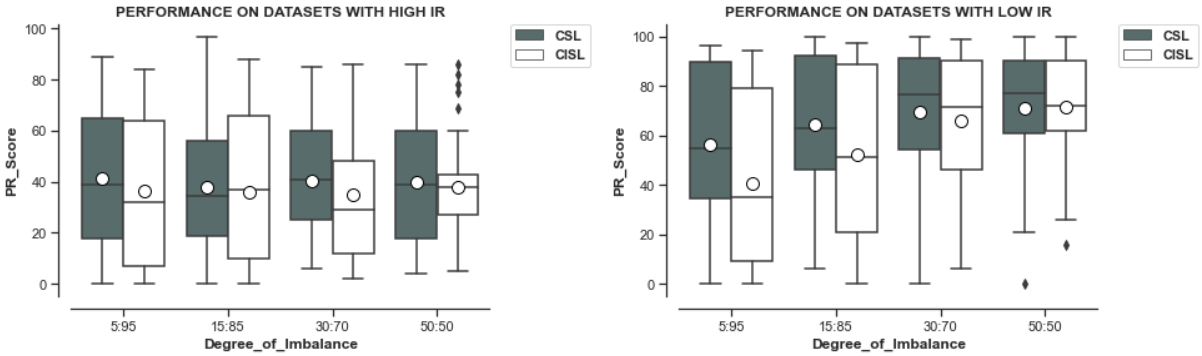
Figure 3.6: PR rank Distribution Tree vs Non-Tree

Another possible viewpoint can be that even though tree-based models performed better than non-tree-based models, their scores are still far from ideal if this can be attributed to data fragmentation, alongside the observation that inducing cost sensitivity did not impact the performance of the tree-based classifiers. The CSL algorithms' ability to handle data fragmentation is neutral. But the validity of this conclusion depends heavily upon the correctness of the assumption that scores obtained by the tree-based classifiers aren't ideal because of the data fragmentation. Another interesting observation is that tree-based and non-tree-based algorithms almost performed equally after inducing cost sensitivity. This aid the case of inducing cost sensitivity into non-tree-based models further.

3.2.4.4 Use Case IV: Low *vs.* High Imbalance-Ratio Datasets

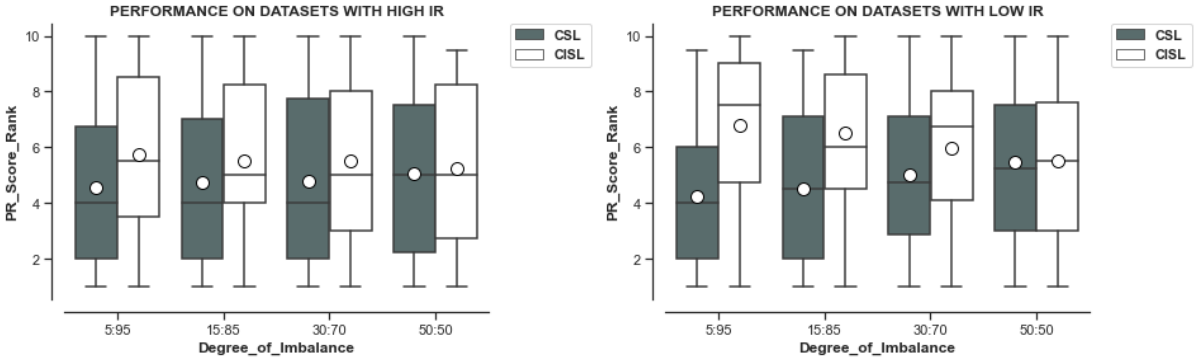
The imbalance ratio determines the number of majority-class samples present for one minority sample. Datasets with High IR have more majority samples for one minority sample and vice versa. CSL algorithms performed better than the CISL algorithms in both aspects at all degrees of imbalance in terms of mean and median except at 15:85 degrees in the low imbalance ratio datasets aspect. We can observe better agreement between the scores of CSL algorithms compared to their CISL algorithms in handling High IR datasets as they have a better interquartile range throughout all the degrees of imbalance. This pattern continued even in the low IR aspect except in the zero-degree imbalance case, where CISL algorithms have comparatively

better distribution. From Fig. 3.7, it is evident that the classifier's performance decreases as the skewness increases. Even the performance of CSL algorithms dropped while moving to low IR datasets from high IR, yet they performed quite well compared to their CISL counterparts. Mienye and Sun reported similar observations in their study on the application of CSL algorithms in medical applications [14].



(a) On Datasets With High IR (b) On Datasets With Low IR

Figure 3.7: PR score Distribution: IR



(a) On Datasets With High IR (b) On Datasets With Low IR

Figure 3.8: PR rank Distribution: IR

An interesting viewpoint in this use case would be to analyse the effectiveness of the balancing method for creating zero-degree imbalance since the use case is divided into low and high IR aspects based on inherent skewness. CISL algorithms could outperform CSL algorithms at zero-degree imbalance degrees, which suggests balancing works better in combination with CISL algorithms than with the other. A stronger pattern supporting the aid of the CSL algorithms can be seen in rank distributions (Fig. 3.8). CSL algorithms could maintain relatively the same mean and median in both aspects. They could even maintain the interquartile ranges and

the ranges of the boxes at the extreme imbalance and at 15:85 case that show consistency and stability of CSL algorithms at a higher degree of imbalance. The difference between means, medians, ranges, and interquartile ranges increased between CSL and CISL algorithms in moving from low to higher IR. These observations also aid the usage of rank distributions based on PR scores as a performance metric for cases of imbalance.

3.2.5 Discussion

The results presented emphasis that CSL algorithms handled most of the mentioned aspects, i.e., medical and non-medical, small and large minority classes, tree-based and non-tree-based, and low and high IR far better than the conventional algorithms. Their performance showed better distributions in terms of means, medians, ranges, and interquartile ranges. With a few notable exceptions, in many aspects, CISL algorithms performed better than CSL algorithms at zero-degree imbalance, which strengthened the argument that inducing cost sensitivity may not contribute much to the balanced distribution. This can also be attributed to the usage of IR for weighting the classes which assigns equal weightage to the classes in the case of the balanced variant. Another is the case of tree-based and non-tree-based models use case, where we could not find conclusive evidence either supporting or discarding the applicability of CSL algorithms ahead of their counterparts in cases of data fragmentation. Yet, that use case provided crucial insights into how various types of classifiers(tree and non-tree based) respond to the induction of cost sensitivity. Most of these observations are strongly represented by the plots of PR ranks than that of the PR scores. This aids the utility of PR rank measure for the assessment of classifiers in cases of imbalance. Some of the plots depicting the scores and ranks exhibit outliers (for example, Fig. 3.7b), likely due to the heterogeneous nature of the algorithms in the group, differing in complexity and learning capability. Such diversity can result in exceptional performance by a particular classifier, causing it to stand out from the rest of the group. Examining these cases in more detail, considering individual algorithms, could offer valuable insights into their behaviour.

3.3 Conclusion

In this chapter, we analyse the effectiveness of CSL algorithms in handling some of the most prominent inherent characteristics of imbalanced data like class separability, data fragmentation, and sample size of the minority class. These factors contribute to the challenges of data imbalance alongside the skewness aspect, which makes the imbalance problem even more challenging and worth investigating. The work presented in this paper aids the application of cost-sensitive algorithms, by understanding the characteristics of class imbalance data, for various real-life applications involving class imbalances. Our methodology includes generating various dataset variants at varying degrees of imbalance. Followed by applying a classifier, i.e., machine learning algorithms in our case, on each of the variants with and without cost-sensitivity. Then calculate both the scores and ranks of the classifiers using the obtained PR curves. We then aggregated the results and analysed the behaviour of CSL algorithms under different predefined use cases. Our use case study shows the supremacy of CSL algorithms in handling various aspects of imbalance in comparison with the conventional CISL counterparts.

The efficacy of our methodology and the empirical analysis regarding the behaviour of CSL algorithms can be continued in many possible directions. This work used Inverse Class Distribution Ratio (ICDR) to assign class weights. A more robust way of assigning class weights, using complexity measures like Tomek Link Complexity Measure (TLCM) [15], etc. can also be investigated, improving the precision of the details regarding the learning complexity of a dataset. This paper analysed only a handful of cost-sensitive machine learning algorithms and their ins and outs in handling the challenges of class imbalance. Future work may include cost-sensitive versions of other machine learning algorithms to generalize the inferences made. They can also extend this argument to understand the behaviour of cost-sensitive deep-learning algorithms. Increasing the number of degrees of imbalance is also an interesting extension of this work, which may enhance the detail of how the performance of a CSL algorithm varies as a function of the degree of imbalance. This study is applied to algorithms dealing with cost sensitivity; future work may assess the behaviour of the other machine learning algorithms for class imbalances. Designing analytically robust use cases can be another direction of future work for enhancing the precision of inferences regarding the characteristics of the algorithms.

4

Evaluation of Cost Matrices for Cost-Sensitive Learners

This chapter offers a comprehensive introduction to cost matrices and their significance in determining the costs associated with misclassifications in different classes for cost-sensitive classifiers. The focus is on analyzing the impact of cost matrices on the performance of the reference model, which is a Cost Sensitive Logistic Regression (CSLR), across varying degrees of imbalanced data and offers insights into how the selection of class weights impacts the performance at different degrees.

4.1 Cost Matrices and their Utility in CSLs

Class imbalance and other data characteristics, such as feature set and class separability, could affect the performance of most machine learning algorithms. This can be attributed to the algorithm's primary assumption about the data being class-balanced and indifferent weightage among different misclassification errors i.e., True Positive, False Positive, etc.

Cost-sensitive learning induces a cost matrix consisting of the weighting scheme for these misclassification errors into the algorithm's training process. This forces the model to penalize the misclassification errors according to the skewness of the data to reduce the learning bias of the model towards the majority class. Unlike conventional algorithms, the motivation for these algorithms is to reduce the overall cost of misclassification.

Cost-Sensitive Learning focuses on the idea that false negatives must be weighed more than false positives during the calculation of error. However, defining an optimized cost matrix can be challenging if not given a prior one [38]. This work aims at understanding the efficiency of cost-sensitive algorithms in dealing with binary classification problems with varying degrees of imbalance. We use different costs for weighing false positives and false negatives, which are provided to the algorithm during the training in class weights. Since, the attention on classes changes with the change in cost values, selecting the right cost matrix is crucial for attaining optimal performance. In this chapter, we aim to verify the impact of varying cost values at various degrees of imbalance. We use the Cost-Sensitive Logistic Regression (CSLR) algorithm as a reference model and empirically evaluate the performance of cost-sensitive algorithms over varying degrees of imbalanced data through the proposed methodology.

The research contribution of this work is the proposed methodology that can be used to empirically verify the behavioural patterns of cost-sensitive algorithms and their applicability over imbalanced data. Our methodology empirically verifies whether the behaviour of cost-sensitive algorithms varies with varying degrees of imbalanced data. In this work, we used class weights as the hyperparameter, with which we verified the behaviour of cost-sensitive algorithms. This hyperparameter can be customized in future works related to cost-sensitive learning algorithms.

4.2 Related Work

This section discusses various works in cost-sensitive learning and identifies the associated research gaps. Entire work in cost-sensitive learning can be categorized into two verticals: designing a cost matrix and modifying existing learners to make them cost-sensitive [38].

4.2.1 Designing of Cost Matrix

Defining a cost matrix is usually the core part of any cost-sensitive algorithm. Typically, prior availability of a cost matrix is suitable, except that it requires a domain expert to validate. Moepya et al. [39] used eight different cost matrices to analyse the effectiveness of cost-sensitive algorithms in financial fraud detection using transaction statements. According to them, the true costs of misclassification were hard to find. Therefore, they used pre-determined cost matrices with different levels of cost items. A more popular approach is to use the inverse class distribution ratio for weighing the misclassification errors, i.e., the weight of FP error = the number of majority samples and vice versa. Mienye et al. [14] used this ratio for analysing the performance of cost-sensitive algorithms to handle imbalanced medical data. This method of setting the cost values may be ineffective as an imbalance may not be the sole reason for the inefficiency of a model [40].

Maloof et al. [37] proposed threshold moving, which shifts the output threshold close to majority class instances, making it hard to misclassify samples with higher costs. It employs cost values at the classification phase rather than the training phase. This method is considered unpopular yet effective as a resampling technique for solving imbalance. Domingos et al. [36] proposed a meta-cost approach for solving the class imbalance. Its underlying principle is to create multiple replicates of the original training data and then employ a classifier on each of them. Then estimate each class's probability based on the votes it received and relabels all the examples with that estimated class. The final step reapplies the classifier on the relabeled training data. In case of any modification to the cost matrix, the algorithm needs to repeat only the final stage of the learning process in contrast to the remaining algorithms, where it needs to repeat the entire learning process.

4.2.2 Modification of Learners

Objective functions of the machine learning algorithms weigh all types of errors equally. Cost items are induced into learners' objective functions to make them cost-sensitive. This also forces the learner to reduce the overall cost instead of the training error. Mienye et al. [14] focused on inducing cost values into the objective functions of the classifiers, namely, Logistic Regression, Random Forest, Decision Tree, and XGBoost. They then used the modified objective functions of these algorithms to classify some of the well-known medical imbalanced data sets. This study showed cost-sensitive random forest to be the most effective among its counterparts in dealing with imbalanced medical data. Sun et al. [10] investigated meta-techniques to classify imbalanced data. They proposed three ways to induce cost values into AdaBoost's weight update formula originating three variants, namely, AdaC1, AdaC2, and AdaC3. They induced cost items outside the exponential function for the first variant, inside the function for the second, and both in and out of the function for the last. Based on the empirical analysis, this paper concluded that the performance of AdaC2 is superior among the variants. Cao et al. [41] presented an approach incorporating a combination of performance metrics, i.e., GMean and AUC, into the Support Vector Machine (SVM) classifier's objective function, making it cost-sensitive. Along with the performance metrics, this work also focused on optimising intrinsic parameters, feature subsets, and cost parameters simultaneously.

Research gaps that can be pointed out from the above literature survey are as follows:

1. Most work focused either on comparing cost-sensitive algorithms to their non-cost-sensitive counterparts or on the effectiveness of a specific cost-sensitive algorithm in handling imbalance. but the general behaviour of the cost-sensitive algorithms is rarely discussed in the works mentioned.

2. response of the cost-sensitive algorithms to various characteristics of the imbalanced data sets, i.e., skewness, data fragmentation, class overlapping, etc. is yet to be explored. This work focuses on the skewness aspect of the imbalanced data.

4.3 Methodology for Assessment of Cost Matrices

This section discusses our proposed methodology to empirically evaluate the behaviour of cost-sensitive algorithms over the varied degree of imbalanced data with various class weights.

4.3.1 Imbalancing Groups

The dataset under study is resampled to produce four imbalanced variations. The first version is characterised by high imbalance, with a minority sample to majority sample ratio close to 5:95; the second and third variants are characterised by moderate class imbalance, with imbalance ratios close to 15:85 and 30:70, respectively. With a ratio close to 50:50, the final variant has very little to no imbalance.

4.3.2 Sampling for Imbalancing

Using Random Under Sampling, we initially create an imbalance in our training data by reducing the number of samples of the minority class and increasing the number of samples of the majority class using the ADASYN oversampling method. As the oversampling methods such as SMOTE and ADASYN cannot generate the exact number of samples required, the imbalance ratios are relatively similar to that of the stated rather than accurate.

4.3.3 Selection of Weights

After setting the imbalance, we created a dictionary of hyper-parameters with different class weights, i.e., 100:1, 10:1, 1:1, 1:10, 1:100, and 1:1000 (refer to Table 4.1). We can include any class weight for the experiment, but a more reasonable class weight should weigh a minority class instance more than the majority.

4.3.4 Cost-Sensitive Logistic Regression (CSLR) as the Reference Model

Cost-Sensitive Logistic Regression (CSLR) is a classification algorithm designed to handle imbalanced datasets by considering different costs associated with misclassification. It extends logistic regression by introducing class weights to reflect the imbalanced nature of the data.

The proposed methodology analyzes the behavioral patterns of cost-sensitive algorithms using CSLR as the reference model. It considers four degrees of imbalanced variants of popular datasets and different levels of class weights as hyperparameters. The behavior of CSLR is analyzed using the mean absolute score and Kappa score error bars. The methodology can be easily extended to analyze other CSL classifiers and their efficiency in dealing with imbalance.

4.3.5 Measures for Performance Assessment

Most prominent metrics, such as accuracy, may fail to accurately indicate a classifier's efficiency in dealing with class imbalance. A classifier that predicts the negative class for every training sample can still achieve a high level of accuracy, representing the classifier as efficient. Therefore, we considered a more suitable metric called *Cohen's Kappa* coefficient for our analysis. This can be calculated from the *Confusion Matrix* using the following equation

$$Kappa = \frac{2 \times (TP \times TN - FP \times FN)}{(TP+FP) \times (FP+TN) + (TP+FN) \times (FN+TN)}$$

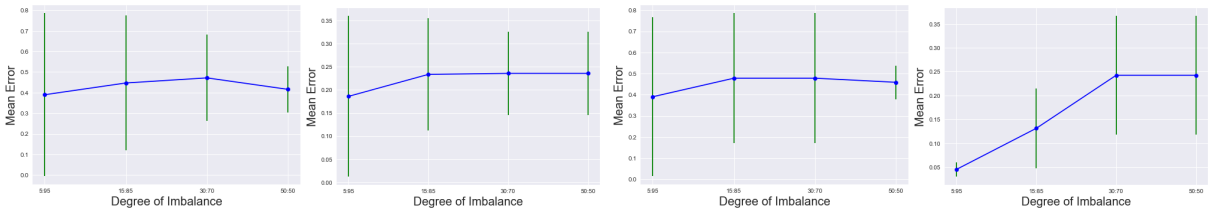
4.4 Results

4.4.1 Experimental Setup

Original versions of the datasets are taken from the UCI repository and loaded into Jupyter Note Book (python 3.6) for the experiment. Entire data sets' are used for creating different degrees of imbalanced data. The standard implementation of ADASYN (from *imblearn*) for resampling, Logistic Regression for model generation, RepeatedStratifiedKFold for creating K folds, i.e., K=10 and GridSearchCV for cross-validation with class weights as hyperparameters are used for the implementation purposes. We instantiated the standard Logistic Regression class present in the sklearn module with the set of class weights mentioned in Table 4.1. We then empirically evaluated the performance of this model over the four imbalanced variants of the four datasets abalone, pima [61], ionosphere and vehicle.

4.4.2 Residual Errors

On every imbalanced variant of the dataset, we calculate the mean and standard deviation of mean absolute error for the seven class weights mentioned. Figure 4.1 illustrate the results using error bars where bars represent the standard deviation of the error; their intercept point represents the mean error over all the weights for a specific variant.



(a) Abalone (b) Vehicle (c) Pima (d) Ionosphere

Figure 4.1: Error distribution for four datsets

4.4.3 Performance Measures

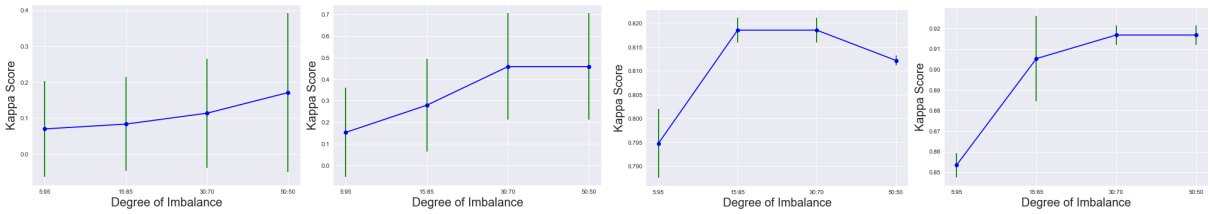
Error bars presented above are based on the mean absolute error, simply the inversion of the accuracy score. As mentioned in section4.4.3, we consider the *Kappa* score, i.e., Cohen's Kappa coefficient, to be a superior metric in representing the performance of a classifier in cases involving imbalance. In this section, we present the performance trends of our Cost-Sensitive Logistic Regression (CSLR) algorithm in terms of the Kappa score.

Table 4.1: Kappa score on the Vehicle dataset

ClassWeigh Degree	100:1	10:1	1:1	1:10	1:100	1:1000	1:10000
5:95	0.000	0.000	0.000	0.619	0.242	0.124	0.087
15:85	0.000	0.000	0.426	0.649	0.365	0.275	0.239
30:70	0.012	0.331	0.858	0.677	0.498	0.437	0.394
50:50	0.012	0.331	0.858	0.677	0.498	0.437	0.394

Table 4.1 shows the kappa score over different class weights and degrees of imbalance for the Vehicle data set. Rows represent different degrees of imbalance, whereas columns represent different class weights, i.e., Minority Class Weight to Majority Class Weight Ratio. Figure 4.2 shows the varying trend of kappa values in the form of error bars, i.e., the error bar represents

standard deviation, and the intercept point represents the mean Kappa score over different class weights.



(a) Abalone (b) Vehicle (c) Pima (d) Ionosphere

Figure 4.2: Kappa Score Distribution for the datasets.

4.4.4 Discussion

Our experiment's primary objective is to analyze the varying performance of cost-sensitive algorithms, for example, the Cost-Sensitive Logistic Regression (CSLR), dealing with different degrees of class imbalance and the impact of varying cost values on the same. Based on the error bars obtained from our experiment, three of the four data sets support our hypothesis, i.e., varying cost values influence the performance of CSLR while dealing with a higher degree of class imbalance But this has little to no impact when it comes to well/almost balanced data. This can be observed in terms of the growing length of error bars, depicting a higher variance of errors, i.e., the unstable performance of the classifier over different class weights. Our experiment on the last data set, i.e., Ionosphere does not show any significant pattern either supporting or contradicting the same. This can be attributed to the data set's unique characteristics, the re-sampling method used to create the variants, and the scaling method used. The classifier's performance on this data set varied when a different re-sampling or scaling method was used during the pre-processing phase.

The means line, i.e., the line joining the mean error over different degrees of imbalance, does not show any significant pattern for data sets. It is neither converging nor diverging over different degrees of imbalance. This can be attributed to the fact that the accuracy score does not represent the cases involving imbalance well, and so is the mean absolute error. This motivated us to provide the results of our experiments in terms of the Kappa score. Table 4.1 shows the Kappa scores obtained on the vehicle dataset. Figure 4.2 represent the Kappa score values in

the form of error bars over all the datasets. The Kappa mean line is steadily increasing as we move from extreme imbalance to that of a well-balanced variant in three of the four datasets. This shows that the Cost-Sensitive Algorithms perform well on a low degree of imbalance in the data and vice-versa. These error bars do not show any significant pattern supporting our hypothesis. Suppose the kappa score metric is considered superior to the mean absolute error. In that case, these error bars show that the impact of the selection of class weights on the performance of cost-sensitive algorithms over different degrees of imbalance is subjected to the characteristics of the data set and cannot be generalized.

4.5 Conclusion

This study proposed a methodology for empirically analyzing the behavioural patterns of cost-sensitive algorithms using different degrees of imbalanced data. We consider the Cost-Sensitive Logistic Regression (CSLR) as the reference model to present our methodology. We consider four degrees of imbalanced variants of four popular datasets for our analysis and different levels of class weights as the hyperparameter. The behaviour of the Cost-Sensitive Logistic Regression (CSLR) is then analyzed with the help of two error bars generated based on the mean absolute score and Kappa score, respectively. The discussion section thoroughly discusses their support and resistance to our hypothesis. We suggest the usage of different hyperparameters, different sampling and scaling techniques, and the inclusion of some more degrees of imbalanced variants of the data sets, comparing the performance of different cost-sensitive algorithms over them as directions for future work. This methodology can also be extended to analyze the behavioural patterns of cost-insensitive classifiers and their efficiency in dealing with the imbalance.

5

ADASYN-based Cost Matrix for Cost Sensitive Classifiers

In this chapter, we propose a novel ADASYN (Adaptive Synthetic Sampling) based complexity measure and extensively evaluated it to showcase its superior performance compared to existing complexity measures in the cost matrix design. First, we investigate using data set complexity measures tailored to assess the complexity of imbalanced datasets in the cost matrix design. Secondly, we analyse the impact of these cost matrices on the performance of the Cost-Sensitive Logistic Regression (CSLR) classifier over five datasets and show the effectiveness of the proposed measure of cost matrix design.

5.1 Dataset Complexity Measures for Imbalanced Data

The difficulty of a classification task with imbalanced class priors is commonly quantified using the Imbalance Ratio (IR). IR is calculated as the ratio between the number of points in the largest majority class and the smallest minority class. It is widely used in the cost matrix design because of its simplicity and interpretability. However, research has shown that IR is a weak estimator of problem complexity as it poorly correlates with classifier performance.

To address this limitation and provide a more accurate assessment of the complexity in class-imbalanced datasets, that can be used in the cost matrix design, we propose a novel ADASYN-based approach for assigning misclassification cost weights to the samples. Our measure estimates the complexity of class imbalance by considering the number of minority class samples that have a neighbourhood dominated by majority class samples. We calculate the number of samples belonging to the opposite class within the K-neighbourhood, which indicates the proximity or overlap of the class distributions.

Developing data complexity measures that accurately capture the full difficulty of imbalanced datasets is crucial for the design of a cost matrix and it requires an understanding of the underlying mechanisms that contribute to complex classification tasks. These complexity measures can assist researchers in identifying challenging imbalanced learning benchmark datasets and understanding which aspects of complexity (such as class overlap, noise, sub-concepts, etc.) combine with a class imbalance to make them difficult learning tasks. Moreover, accurate measures of imbalance complexity can be valuable tools in meta-learning and can support practitioners in determining potential pitfalls in a given dataset, suggesting appropriate preprocessing steps, and guiding the selection of classification algorithms besides the cost matrix design.

5.2 Review of Existing Complexity Measures

N1-Measure [65] The N1 complexity measure is designed to assess the complexity of imbalanced datasets by considering the presence of differently labelled neighbours for instances of a target class. It utilizes a k-nearest neighbours (kNN) graph and a minimum spanning tree

(MST) to capture local relationships and connections between instances. The measure focuses on immediate neighbours within the MST, which may limit its ability to capture global dependencies. The choice of the dissimilarity metric used in constructing the kNN graph should align with the dataset characteristics. For continuous data, Euclidean distance is commonly used, while Gower distance accommodates mixed data types. The N1 complexity measure provides insights into the severity of complexity in imbalanced datasets, helping researchers identify challenging datasets and select appropriate preprocessing steps or classification algorithms. However, it is important to be aware of its limitations and interpret the results accordingly.

N3-Measure [65] The N3 complexity measure focuses on evaluating the presence of differently labelled neighbours among the three nearest neighbours of instances in the target class. By examining this slightly larger neighbourhood, the N3 measure provides insights into local relationships, potential class overlap, and complex decision boundaries. One of the characteristics of the N3 measure is its expanded scope compared to the N1 measure. The N1 complexity measure considers only the immediate nearest neighbour of instances in the target class. In contrast, the N3 measure takes into account the three nearest neighbours, providing a broader perspective on the local relationships and capturing more information about the dataset's complexity.

By considering a larger neighbourhood, the N3 measure can uncover instances that may be close to decision boundaries or regions where different classes coexist. It offers a more comprehensive evaluation of the complexity, highlighting the challenges posed by class overlap and ambiguity. On the other hand, the N1 complexity measure focuses solely on the immediate nearest neighbour. While it provides a measure of local complexity, it may not capture the full extent of the complexity present in the dataset. It could overlook instances that are further away but still contribute to the overall complexity of the imbalanced dataset.

Tomek Links Complexity Measure (TLCM) [15] The Tomek Links Complexity Measure (TLCM) calculates the proportion of Tomek links involving instances from the minority class in a dataset. Tomek links represent pairs of instances from different classes that are each other's nearest neighbours, indicating potential classification errors. The TLCM measure quantifies the

number of Tomek links involving instances from the minority class and divides it by the total number of instances in the minority class. This provides a measure of the proportion of Tomek links, indicating the complexity of class overlap and proximity to the decision boundary.

Compared to the N1 and N3 complexity measures, the TLCM measure specifically focuses on Tomek links involving the minority class instances. It highlights the impact of these links on classification accuracy and provides insights into the challenges posed by class overlap. Overall, the TLCM measure offers a targeted evaluation of complexity based on Tomek links, while the N1 and N3 measures provide a more general assessment of complexity based on nearest-neighbour relationships.

Imbalance Ratio [14] The Imbalance Ratio (IR) is a measure used to quantify the class imbalance in a dataset. It is calculated by dividing the number of instances in the majority class by the number of instances in the minority class. The Imbalance Ratio provides a simple and interpretable measure of class imbalance in a dataset. A higher IR value indicates a greater imbalance between the classes, with the majority class being significantly larger than the minority class.

The Imbalance Ratio (IR) and the complexity measures (N1, N3, and TLCM) provide different perspectives on imbalanced datasets. The IR measure focuses on quantifying the class imbalance by calculating the ratio between the number of instances in the majority class and the number of instances in the minority class. It offers a simple and straightforward indication of the disparity between the classes based on sample counts.

On the other hand, the complexity measures (N1, N3, and TLCM) delve deeper into the specific challenges posed by imbalanced datasets. They consider local relationships, class overlap, and proximity to decision boundaries to assess the complexity associated with class imbalance. While the IR measure provides an overall view of class imbalance, the complexity measures offer more nuanced insights into the characteristics of the imbalanced dataset. They capture the presence of differently labelled neighbours (N1 and N3) or the existence of Tomek links (TLCM), highlighting regions of class overlap and complexity in the decision boundaries, this aids their utilization for the design of cost matrices.

5.3 Proposed ADASYN-based Sample Weighting Method

Algorithm 1: Calculate Normalized Costs

Input : train_data: The training data samples

train_label: The corresponding labels for the training data

k: The number of nearest neighbors to consider

Output: normalized_costs: The normalized cost values for samples 1 to n

```
train_data, train_label, k;
num_samples_class1 ← count(train_label, 1);
num_samples_class0 ← count(train_label, 0);
cost_values ← {};
normalized_costs ← {};
for i ← 1 to n do
    k_nearest_neighbors ← find_k_nearest_neighbors(train_data, train_data[i],
      k);
    count_class0 ← 0;
    count_class1 ← 0;
    for j ← 1 to length(k_nearest_neighbors) do
        count_class0 ← count_class0 + (train_label[j] = 0);
        count_class1 ← count_class1 + (train_label[j] = 1);
    if train_label[i] = 0 then
        cost_values ← cost_values ∪ {count_class1 + 1 × num_samples_class1};
    else
        cost_values ← cost_values ∪ {count_class0 + 1 × num_samples_class0};
sum_cost_values ← sum(cost_values);
for cost_value in cost_values do
    normalized_costs ← normalized_costs ∪ {cost_value / sum_cost_values};
return normalized_costs;
```

The algorithm presented calculates the normalized costs for each sample in a given training dataset, considering the class imbalance. The normalized costs measure the impact of misclassification for each sample and can be used to account for the class distribution during model training or evaluation.

The algorithm takes as input the training data samples, their corresponding labels, and the number of nearest neighbors (k) to consider. It initializes variables to store the number of samples in each class (class 1 and class 0), an empty list for the cost values, and another empty list for the normalized costs. It iterates over each sample in the dataset. For each sample, it finds the k nearest neighbors using the given distance metric. It then counts the number of samples from class 0 and class 1 among the nearest neighbors.

Depending on the label of the current sample, the algorithm calculates the cost value. If the sample belongs to class 0, the cost value is calculated as the count of class 1 neighbors plus 1, multiplied by the number of samples in class 1. Similarly, if the sample belongs to class 1, the cost value is calculated as the count of class 0 neighbors plus 1, multiplied by the number of samples in class 0. The cost values are stored in the list "cost_values." After iterating through all samples, the algorithm calculates the sum of all cost values. Then, it calculates the normalized costs by dividing each cost value by the sum of all cost values. The normalized costs represent the relative impact of misclassification for each sample, considering the class distribution. Finally, the algorithm returns the list of normalized costs.

This measure allows assigning higher weights to samples from the minority class, reflecting the higher cost of misclassifying such samples in imbalanced datasets. By incorporating the normalized costs into the learning process, models can be encouraged to prioritize the correct classification of minority class samples, leading to improved performance on imbalanced data.

5.4 Methodology for Performance Assessment

5.4.1 Cost Matrix Construction

Most studies on cost-sensitive classification commonly utilize the Inverse Class Distribution Ratio (ICDR) as a class weight, which effectively captures the data imbalance by considering

the number of samples from each class. However, IR alone fails to account for the complexity associated with overlap or distribution at the sample level. Consequently, it proves to be inadequate in fully quantifying the complexity of the imbalanced data. In our approach, we introduce complexity measures that provide a more comprehensive assessment of the imbalance problem by incorporating the neighborhood factor of the samples. By calculating the dataset's complexity using these measures, we derive a corresponding cost matrix, which is then incorporated into the classifier during training. This approach ensures that the classifier is trained with a more nuanced understanding of the dataset's complexity, leading to improved performance in handling class imbalance.

5.4.2 Model Training

In this study, we utilize the standard version of logistic regression available in the sklearn module as a reference model for analysis. To implement our proposed methodology, we incorporate our measure, which calculates the cost at the sample level, by providing it as the sample weight while fitting the training data. Additionally, we incorporate the remaining complexity measures i.e. N1, N3, and TLCM, by providing their calculated values for each dataset to the model as class weights. We opt for default values for all other hyperparameters available in the Logistic Regression's API to minimize variability and ensure a fair comparison between the methods.

5.4.3 Measures for Performance Assessment

We have considered various performance metrics to draw meaningful conclusions regarding the assessment of imbalanced cases. Accuracy has provided an overall measure of correctness in predictions, while precision and recall have shed light on the model's ability to correctly identify positive instances. The F1-score, being a balance of precision and recall, has been effective in evaluating classification models in imbalanced datasets. Kappa has served as a robust evaluation metric by measuring the agreement between predicted and actual classifications. Additionally, GMean has captured the performance on both positive and negative instances, making it suitable for imbalanced datasets. Lastly, the AUC has been useful in assessing the model's ability to rank instances correctly, particularly in imbalanced class distributions. Moreover,

we have leveraged the Precision-Recall (PR) score to analyze the trade-off between precision and recall, enabling a comprehensive evaluation of the model's accuracy in identifying positive instances within the context of our experiment.

5.5 Results and Discussion

5.5.1 Data sets and Pre-processing

In our empirical analysis, we utilized a diverse set of five datasets obtained from various sources, including the UCI repository [1] and the Pima dataset from [61]. These datasets predominantly consisted of binary classification problems, and for multi-class datasets, we converted them into binary by considering the class with the highest sample count as the majority and the class with the lowest sample count as the minority. Further information about the datasets can be found in Table 3.1.1 and Table 3.2. It is worth noting that most of the datasets contained missing values, which were imputed using the KNNImputer from the sklearn library. Additionally, to ensure consistency, we applied the MinMaxScaler from sklearn to scale the values within the datasets. Each dataset was divided into two subsets, with fifteen percent of the samples reserved for testing, while the remaining samples were used for training and the calculation of complexity measures. Following the training phase, we evaluated the efficiency and performance of the models by testing them on the initially set-aside test data.

5.5.2 Performance Assessment

[1] https://archive-beta.ics.uci.edu/

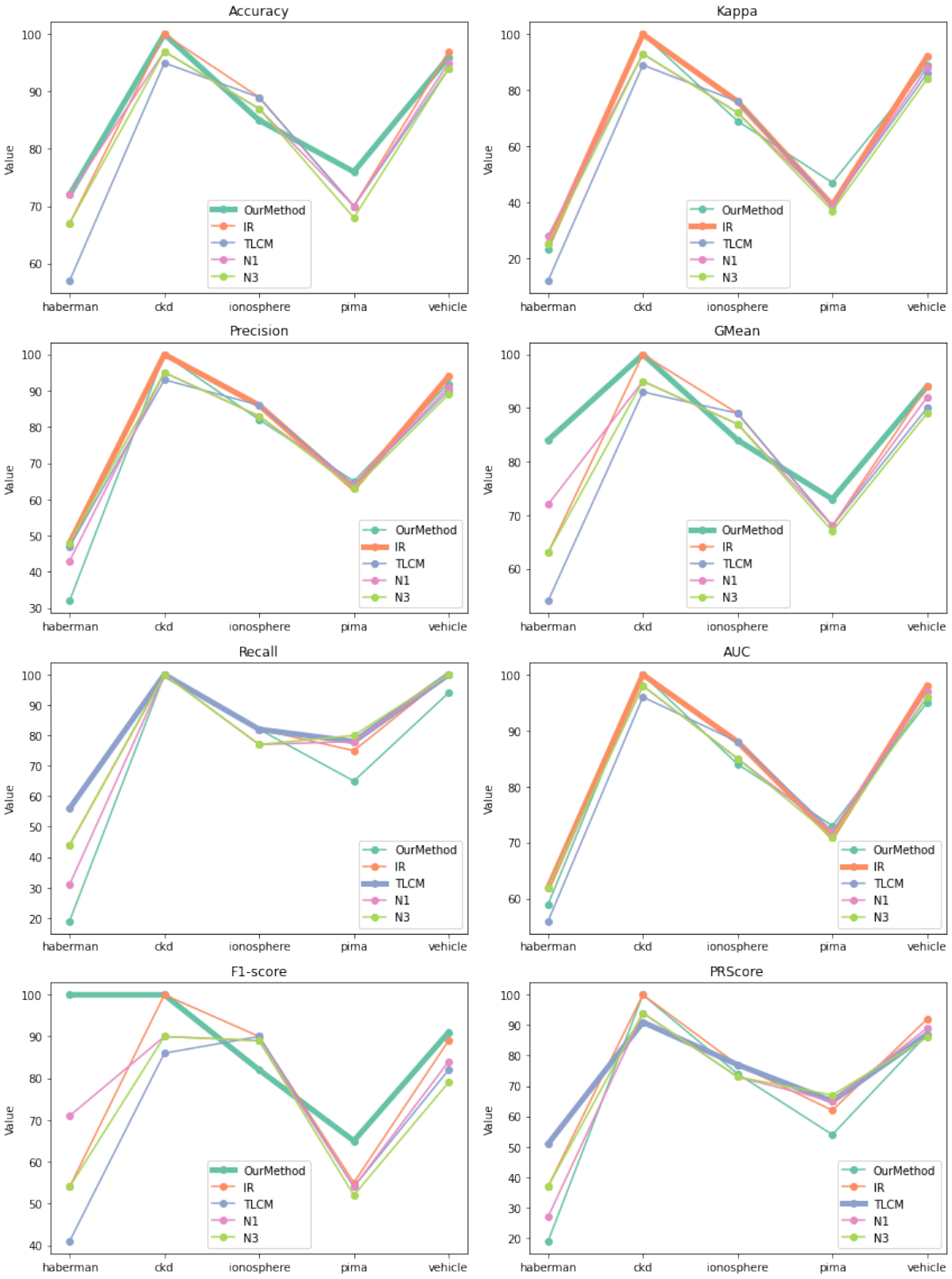


Figure 5.1: Performance Overview

5.5.3 Discussion

The proposed measure consistently achieves high performance across multiple datasets and metrics, including accuracy, precision, recall, F1-score, Kappa, GMean, AUC, and PR Score. These results indicate its effectiveness in handling class imbalance and accurately classifying instances. Specifically, the measure demonstrates high accuracy and precision scores, indicating its ability to make correct predictions and minimize false positives. The high recall score suggests a low likelihood of missing positive instances. The balanced F1 score showcases a good trade-off between precision and recall. The high Kappa score indicates significant agreement between predicted and actual classifications, while the high GMean score reflects its effectiveness in both positive and negative instances. The high AUC score demonstrates the method's capability to rank instances correctly, making it well-suited for imbalanced datasets.

Moreover, the competitive PR Score achieved by the proposed measure reinforces its proficiency in balancing precision and recall, especially when focusing on positive instances. This score highlights its ability to achieve high precision while maintaining a reasonable level of recall, which is crucial in imbalanced datasets. In comparison, other measures like N1, N3, IR, and TLCM exhibit varying levels of performance across metrics and datasets. N1 shows high accuracy but lower precision and recall, indicating potential challenges in correctly identifying positive instances and avoiding false positives or false negatives. N3 performs similarly to N1 in accuracy but exhibits higher precision and recall, suggesting its effectiveness in classifying positive instances and reducing false positives and false negatives.

IR performs well in accuracy, precision, recall, and F1-score but relatively lower in Kappa and AUC, suggesting limitations in capturing agreement beyond chance and discriminating between classes. TLCM demonstrates a balance between accuracy, precision, and recall, with high recall scores indicating the correct identification of positive instances. However, its precision scores are comparatively lower, indicating challenges in avoiding false positives. In summary, the proposed measure consistently outperforms the other measures across various metrics, showcasing its robustness in classification tasks, particularly in the presence of class imbalance. Its high accuracy, precision, recall, F1-score, Kappa, GMean, AUC, and PR Score collectively indicate its reliability, effectiveness, and ability to balance precision and recall in

imbalanced datasets.

5.6 Conclusion

The proposed measure, ADASYN (Adaptive Synthetic Sampling)-based complexity measure, has been extensively evaluated across multiple datasets and performance metrics, including accuracy, precision, recall, F1-score, Kappa, GMean, AUC, and PR Score. The consistently high performance achieved by the proposed measure demonstrates its effectiveness in handling class imbalance and accurately classifying instances. The results indicate that the proposed measure excels in various aspects.

Its high accuracy and precision scores highlight its ability to make correct predictions and minimize false positives. The high recall score suggests a low likelihood of missing positive instances. The balanced F1 score showcases a good trade-off between precision and recall. Furthermore, the high Kappa score indicates significant agreement between predicted and actual classifications, while the high GMean score reflects its effectiveness in both positive and negative instances. The high AUC score demonstrates the measure's capability to rank instances correctly, making it well-suited for imbalanced datasets. Additionally, the competitive PR Score achieved by the proposed measure reinforces its proficiency in balancing precision and recall, particularly for positive instances. This score highlights its ability to achieve high precision while maintaining a reasonable level of recall, which is crucial in imbalanced datasets.

In comparison, other measures, such as N1, N3, IR, and TLCM, exhibit varying levels of performance across metrics and datasets. While some of these measures demonstrate strengths in certain aspects, they generally fall short of achieving the consistently high performance demonstrated by the proposed measure. In summary, the results of our study confirm that the proposed ADASYN-based complexity measure consistently outperforms other measures across multiple datasets and metrics. Its robustness in classification tasks, particularly in the presence of class imbalance, is evidenced by its high accuracy, precision, recall, F1-score, Kappa, GMean, AUC, and PR Score. These findings establish the reliability, effectiveness, and ability of the proposed measure to balance precision and recall in imbalanced datasets.

6

Conclusion and Future Work

6.1 Work Summary

Throughout this dissertation, we have addressed the challenges posed by class imbalance in classification tasks and investigated the effectiveness of cost-sensitive learning (CSL) algorithms. We have focused on understanding the intrinsic characteristics of imbalanced data and evaluated the performance of CSL algorithms across various types of classifiers, different data domains, and varying degrees of class imbalance.

In Chapter 2, we provided an in-depth analysis of the class imbalance problem, reviewed existing literature on solutions, and introduced cost-sensitive learning algorithms. We discussed the definition of class imbalance, explored data-level and algorithmic-level methods to handle imbalance, and presented performance metrics for evaluating imbalanced cases. In Chapter

3, we conducted an empirical analysis of cost-sensitive learning algorithms to assess their effectiveness in addressing the intrinsic characteristics of imbalanced data. We evaluated the performance of cost-sensitive (CSL) and cost-insensitive (CISL) algorithms across varying degrees of imbalance using fifteen datasets from diverse domains. The results provided insights into the performance of CSL algorithms and their superiority in handling class imbalance.

Chapter 4 focused on understanding the significance of cost matrices in cost-sensitive classifiers. We investigated the impact of cost matrices on the performance of the reference model, Cost-Sensitive Logistic Regression (CSLR), across different degrees of imbalanced data. By examining the selection of class weights, we gained insights into how these choices influenced the classifier's performance. In Chapter 5, we proposed a novel complexity measure based on Adaptive Synthetic Sampling (ADASYN) and evaluated its effectiveness in a cost matrix design. We explored the utilization of data set complexity measures tailored to assess the complexity of imbalanced datasets and analyzed the impact of these cost matrices on the performance of the CSLR classifier using five datasets. The results demonstrated the effectiveness of the proposed ADASYN-based complexity measure in a cost matrix design.

Lastly, in Chapter 6, we concluded the dissertation by summarizing the research contributions and discussing future directions. We emphasized the importance of handling class imbalance, highlighted the effectiveness of CSL algorithms, and provided insights into best practices for cost-sensitive learning. Overall, this dissertation contributes to the understanding of class imbalance and the effectiveness of cost-sensitive learning algorithms. The research findings and insights gained from the empirical analysis and evaluation of different approaches can guide researchers and practitioners in selecting appropriate techniques and designing cost matrices to improve classification performance in imbalanced datasets.

6.2 Research Contributions

The research work presented in this study has made several significant contributions to the field of handling class imbalance and cost-sensitive learning. The key research contributions can be summarized as follows:

Proposal of ADASYN-based Complexity Measure: The study introduces a novel complexity measure based on Adaptive Synthetic Sampling (ADASYN). This proposed measure provides insights into the characteristics of imbalanced datasets, considering factors such as class separability and sample size of the minority class. The development of this measure expands the existing knowledge on complexity measures for cost-sensitive learning and offers a new approach to guide cost matrix design.

Empirical Analysis of CSL Algorithms: This study conducted an extensive empirical analysis to evaluate the effectiveness of cost-sensitive learning (CSL) algorithms in handling the intrinsic characteristics of imbalanced data. Through a predefined set of use cases and performance evaluations across multiple datasets and metrics, the study demonstrates the superiority of CSL algorithms in addressing class imbalance compared to conventional cost-insensitive algorithms.

Future Directions and Scope: The research outlines several future directions and scope for further exploration in the field. These include investigating different approaches for assigning class weights, evaluating the behaviour of cost-sensitive algorithms with varying degrees of imbalance, and conducting real-world case studies to validate their practical utility.

Overall, the research work presented in this study significantly contributes to the understanding and advancement of cost-sensitive learning in handling class imbalance. The proposed ADASYN-based complexity measure, empirical analysis of CSL algorithms, evaluation of performance metrics, and future directions outlined in the study collectively contribute to the knowledge base in the field and provide a foundation for further research and practical applications.

6.3 Directions for Future Work

Exploring Robust Class Weight Assignment: In this study, class weights were assigned using the Inverse Class Distribution Ratio (ICDR). Future work can investigate more robust ways of assigning class weights, such as incorporating complexity measures like the Tomek

Link Complexity Measure (TLCM) or other relevant measures. This exploration can improve the precision and accuracy in capturing the learning complexity of a dataset.

Generalizing to Other Algorithms and Domains: This study focused on a select set of cost-sensitive machine learning algorithms. Future research can include cost-sensitive versions of other machine learning algorithms to generalize the findings. Additionally, extending the analysis to cost-sensitive deep learning algorithms would provide insights into their behaviour in handling class imbalance. This exploration can enhance the understanding of the effectiveness of a broader range of algorithms in imbalanced classification tasks.

Increasing Degrees of Imbalance: The current study considered a limited number of degrees of imbalance. Future work can expand the analysis by including a wider range of degrees of imbalance. This extension will provide a more detailed understanding of how the performance of cost-sensitive learning algorithms varies as a function of the degree of imbalance. It can help identify the optimal degree of imbalance at which cost-sensitive algorithms exhibit the best performance.

Designing Analytically Robust Use Cases: Future work can focus on designing analytically robust use cases to enhance the precision of inferences regarding algorithm characteristics. This can involve considering additional factors such as class overlapping, dataset complexity, and other relevant characteristics quantified in a robust manner. These use cases will provide a more comprehensive understanding of algorithm behaviour and performance in handling class imbalance.

Extending the Methodology to Cost-Insensitive Classifiers: The methodology proposed in this study can be extended to analyze the behavioural patterns of cost-insensitive classifiers as well. Comparing the efficiency of cost-sensitive and cost-insensitive classifiers in dealing with class imbalance will provide a comprehensive understanding of the advantages and limitations of both approaches.

Generalization and Validation: The current evaluation of the proposed measure has been conducted across multiple datasets and performance metrics. To further validate its effectiveness, future research can expand the evaluation to include a more diverse range of datasets from various domains. This will help assess the generalizability of the measure and its performance in different application scenarios.

Extension to Different Cost-Sensitive Algorithms: The proposed ADASYN-based complexity measure has been evaluated in the context of cost matrix design for the Cost-Sensitive Logistic Regression (CSLR) classifier. Future research can explore its applicability to other cost-sensitive learning algorithms. This extension will help determine the effectiveness of the measure in guiding cost matrix design for a broader range of classifiers and facilitate the selection of suitable cost-sensitive algorithms based on dataset characteristics.

By pursuing these future directions, researchers can further enhance the understanding of cost-sensitive learning algorithms, their behaviour in handling class imbalances, and their applicability to various real-life applications involving imbalanced datasets. Understanding the applicability of the proposed ADASYN-based complexity measure will contribute to the development of more effective solutions for handling class imbalance and facilitate the adoption of appropriate cost-sensitive algorithms in various domains.

List of Publications

In Conference Proceedings

1. Sai Teja Tangudu and Rajeev Kumar. Analysis of cost-sensitive algorithms for degree of imbalancing. In Proc. 6th Int. Conf. Computational Intelligence in Data Science, 2023. Springer. {*In Press*}

References

[1] Haoke Zhang, Hongyi, Sandeep Pirbhulal, Wanqing Wu, and Victor Hugo C De Albuquerque. Active balancing mechanism for imbalanced medical data in deep learning–based classification models. *ACM Trans. Multimedia Computing*, 16(1s):1–15, 2020.

[2] Vaishnavi Nath Dornadula and Sa Geetha. Credit card fraud detection using machine learning algorithms. *Procedia Computer Science*, 165:631–641, 2019.

[3] Degang Sun, Zhengrong Wu, Yan Wang, Qiujian Lv, and Bo Hu. Risk prediction for imbalanced data in cyber security: a siamese network-based deep learning classification framework. In *Proc. Int. Joint Conf. Neural Networks*, pages 1–8. IEEE, 2019.

[4] Shigang Liu, Yu Wang, Jun Zhang, Chao Chen, and Yang Xiang. Addressing the class imbalance problem in twitter spam detection using ensemble learning. *Computers & Security*, 69:35–49, 2017.

[5] R Kumar, Wan-Ching Chen, and Peter Rockett. Bayesian labelling of image corner features using a grey-level corner model with a bootstrapped modular neural network. In *Proc. 5th Int. Conf. Artificial Neural Networks (Conf. Publ. No. 440)*, pages 82–87. IET, 1997.

[6] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proc. 24th Int. Conf. Machine Learning*, pages 935–942, 2007.

[7] Firuz Kamalov, Amir F Atiya, and Dina Elreedy. Partial resampling of imbalanced data. *arXiv preprint arXiv:2207.04631*, 2022.

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[9] Mohammed Temraz and Mark T Keane. Solving the class imbalance problem using a counterfactual method for data augmentation. *Machine Learning with Applications*, 9:100375, 2022.

[10] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[11] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *Advances in Artificial Intelligence: Proc. 3rd Mexican Int. Conf. Artificial Intelligence (MICAI)*, pages 312–321. Springer, 2004.

[12] Nathalie Japkowicz. Class imbalances: are we focusing on the right issue. In *Proc. Workshop Learning from Imbalanced Data Sets II*, volume 1723, page 63, 2003.

[13] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proc. IEEE Int. Joint Conf. Neural Networks (IEEE World Congress Computational Intelligence)*, pages 1322–1328. IEEE, 2008.

[14] Ibomoiye Domor Mienye and Yanxia Sun. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, 25:100690, 2021.

[15] Jonatan Moller Nuutinen Gottcke, Colin Bellinger, Paula Branco, and Arthur Zimek. An interpretable measure of dataset complexity for imbalanced classification problems. In *Proc. SIAM Int. Conf. Data Mining (SDM)*, pages 253–261. SIAM, 2023.

[16] Akanksha Mukhriya and Rajeev Kumar. Building outlier detection ensembles by selective parameterization of heterogeneous methods. *Pattern Recognition Letters*, 146:126–133, 2021.

[17] David F Williamson, Robert A Parker, and Juliette S Kendrick. The box plot: a simple visual method to interpret data. *Annals of Internal Medicine*, 110(11):916–921, 1989.

[18] Anish Sharma and Rajeev Kumar. Imbalanced learning of regular grammar for DFA extraction from LSTM architecture. In *Proc. 11th Int. Conf. Soft Computing for Problem Solving (SocProS)*, pages 85–95. Springer, 2023.

[19] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[20] Gary M Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

[21] Nathalie Japkowicz. Concept-learning in the presence of between-class and within-class imbalances. In *Advances in Artificial Intelligence: Proc. 14th Biennial Conf. Canadian Society for Computational Studies of Intelligence*, pages 67–77. Springer, 2001.

[22] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[23] Gary M Weiss and Foster Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal Artificial Intelligence Research*, 19:315–354, 2003.

[24] Pooja Singh and Rajeev Kumar. Assessing imbalanced datasets in binary classifiers. In *Proc. 11th Int. Conf. Soft Computing for Problem Solving (SocProS)*, pages 291–303. Springer, 2023.

[25] Aida Ali, Siti Mariyam Shamsuddin, and Anca L Ralescu. Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3), 2013.

[26] Ying-Jin Cui, S Davis, Chao-Kun Cheng, and Xue Bai. A study of sample size with neural network. In *Proc. Int. Conf. Machine Learning & Cybernetics*, volume 6, pages 3444–3448. IEEE, 2004.

[27] Giang Hoang Nguyen, Abdesselam Bouzerdoum, and Son Lam Phung. Learning pattern classification tasks with imbalanced data sets. *Pattern Recognition*, pages 193–208, 2009.

[28] Hongbo Shi, Ying Zhang, Yuwen Chen, Suqin Ji, and Yuanxiang Dong. Resampling algorithms based on sample concatenation for imbalance learning. *Knowledge-Based Systems*, 245:108592, 2022.

[29] Pattaramon Vuttipittayamongkol, Eyad Elyan, and Andrei Petrovski. On the class overlap problem in imbalanced data classification. *Knowledge-based Systems*, 212:106631, 2021.

[30] Jerome Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. *Proc. AAAI*, 1, 09 1997.

[31] Sofia Visa and Anca Ralescu. The effect of imbalanced data class distribution on fuzzy classifiers-experimental study. In *Proc. 14th IEEE Int. Conf. Fuzzy Systems*, pages 749–754. IEEE, 2005.

[32] K Ruwani M Fernando and Chris P Tsokos. Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE Trans. Neural Networks & Learning Systems*, 33(7):2940–2951, 2021.

[33] Bronislav Yasinnik. Imbalanced classification via explicit gradient learning from augmented data. *arXiv preprint arXiv:2202.10550*, 2022.

[34] Jonathan Burez and Dirk Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[35] Nuno Moniz and Hugo Monteiro. No free lunch in imbalanced learning. *Knowledge-Based Systems*, 227:107222, 2021.

[36] Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proc. 5th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, pages 155–164, 1999.

[37] Marcus A Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proc. ICML Workshop Learning from Imbalanced Data Sets II*, volume 2, pages 2–1, 2003.

[38] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. Cost-sensitive learning. In *Learning from Imbalanced Data Sets*, pages 63–78. Springer, 2018.

[39] Stephen O Moepya, Sharat S Akhoury, and Fulufhelo V Nelwamondo. Applying cost-sensitive classification for financial fraud detection under high-class imbalance. In *Proc. IEEE Int. Conf. Data Mining Workshop*, pages 183–192. IEEE, 2014.

[40] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[41] Peng Cao, Dazhe Zhao, and Osmar Zaiane. An optimized cost-sensitive svm for imbalanced data learning. In *Proc. Pacific-Asia Conf. Knowledge Discovery & Data Mining*, pages 280–292. Springer, 2013.

[42] Nathalie Japkowicz. Assessment metrics for imbalanced learning. *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 187–206, 2013.

[43] Qiong Gu, Li Zhu, and Zhihua Cai. Evaluation measures of the classification performance of imbalanced data sets. In *Proc. 4th Int. Symp. Computational Intelligence & Intelligent Systems (ISICA)*, pages 461–471. Springer, 2009.

[44] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

[45] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195–215, 1998.

[46] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proc. 23rd Int. Conf. Machine Learning*, pages 233–240, 2006.

[47] Geoffrey I Webb and Kai Ming Ting. On the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, 58:25–32, 2005.

[48] Tom Fawcett and Peter A Flach. A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning*, 58(1):33–38, 2005.

[49] Mike Wasikowski and Xue-wen Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Trans. Knowledge & Data Engineering*, 22(10):1388–1400, 2009.

[50] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Trans. Knowledge & Data Engineering*, 21(9):1263–1284, 2009.

[51] AP Bradley, RPW Duin, P Paclik, and TCW Landgrebe. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In *Proc. 18th Int. Conf. Pattern Recognition*, volume 4, pages 123–127. IEEE, 2006.

[52] John T Hancock, Taghi M Khoshgoftaar, and Justin M Johnson. Evaluating classifier performance with highly imbalanced big data. *Journal of Big Data*, 10(1):1–31, 2023.

[53] Razvan Bunescu, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, 2005.

[54] Jesse Davis, Elizabeth S Burnside, Inês de Castro Dutra, David Page, Raghu Ramakrishnan, Vitor Santos Costa, and Jude W Shavlik. View learning for statistical relational learning: With an application to mammography. In *Proc. IJCAI*, pages 677–683, 2005.

[55] Parag Singla and Pedro Domingos. Discriminative training of Markov logic networks. In *Proc. AAAI*, volume 5, pages 868–873, 2005.

[56] Insurance data set. Data Source: `https://www.kaggle.com/datasets/anmolkumar/health-insurance-cross-sell-prediction`.

[57] Credit fraud data set. Data Source: `www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets`.

[58] Mammography data set. Data Source: `https://www.openml.org/search?type=data&status=active&id=310`.

[59] Portoseguro data set. Data Source: `https://www.kaggle.com/competitions/porto-seguro-safe-driver-prediction`.

[60] Phenome data set. Data Source: `https://www.openml.org/search?type=data&status=active&id=1489`.

[61] Pima data set. Data Source: `https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database`.

[62] Ecg data set. Data Source: `http://www.timeseriesclassification.com/description.php?Dataset=ECG5000`.

[63] Ninapro data set. Data Source: `http://ninapro.hevs.ch/data2`.

[64] D. Rindskopf and M. Shiyko. Measures of dispersion, skewness and kurtosis. In Penelope Peterson, Eva Baker, and Barry McGaw, editors, *Int. Encyclopedia of Education*, pages 267–273. Elsevier, Oxford, third edition, 2010.

[65] Victor H Barella, Luis PF Garcia, Marcilio CP de Souto, Ana C Lorena, and André CPLF de Carvalho. Assessing the data complexity of imbalanced datasets. *Information Sciences*, 553:83–109, 2021.