On imbalanced data and text classification

Aleksi Avela

Aalto University

Aalto University publication series
Doctoral Theses 242/2025

On imbalanced data and text classification

Aleksi Avela

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall A123/A1 (Otakaari 1) on 5 December 2025 at 12:00.

Aalto University
School of Science
Department of Mathematics and Systems Analysis

Supervising professor
Professor Pauliina Ilmonen, Aalto University, Finland

Preliminary examiners
Professor Thomas Verdebout, Université Libre de Bruxelles, Belgium
Professor Jukka Lempa, University of Turku, Finland

Opponent
Professor Thomas Verdebout, Université Libre de Bruxelles, Belgium

Aalto University publication series
Doctoral Theses 242/2025

ISBN 978-952-64-2863-5 (paperback)
ISBN 978-952-64-2862-8 (pdf)
ISSN 1799-4934 (print)
ISSN 1799-4942 (pdf)

http://urn.fi/URN:ISBN:978-952-64-2862-8

PunaMusta Oy
Helsinki 2025

Aalto University

ABSTRACT

Author Aleksi Avela

Name of the doctoral thesis On Imbalanced Data and Text Classification

Article-based thesis

Number of pages 106

Keywords classification, imbalanced data, text classification

The vast amounts of information available today call for smart ways to analyze and make decisions based on data. One of the most prominent approaches is machine learning, that is, algorithms which utilize data for discovering patterns and learning to make optimal decisions. This thesis focuses on one important category of machine learning: classification, in which the aim is to learn rules that can be used for predicting the classes or labels of observations.

On top of classification in general, this thesis considers two sub-problems of it – both separately and mixed together – which are imbalanced data and text classification. Imbalanced data refers to classification tasks where one or some of the classes are notably rarer compared to the other class(es). Observations belonging to a rare class are typically the ones that have a high value, but, without modifications, many classification algorithms struggle with finding these rare observations. Text classification refers to applying classification algorithms to tasks involving natural language documents.

The thesis includes an introduction to classification and the analysis of text data and three publications. The first publication presents an application of text classification for measuring the economic sentiment in Finland based on news titles. The second publication considers imbalanced data and text data together and introduces a new method for addressing both challenges simultaneously. The third publication discusses the – perhaps surprisingly challenging – question of how different classifiers should be evaluated and compared when dealing with imbalanced data.

Aalto-yliopisto

TIIVISTELMÄ

Tekijä Aleksi Avela

Väitöskirjan nimi Epätasaisesta datasta ja tekstin luokittelusta

Artikkeliväitöskirja

Sivumäärä 106

Avainsanat luokittelu, epätasainen data, tekstin luokittelu

Nykyään on saatavilla paljon informaatiota, ja siksi on tärkeää kehittää menetelmiä, joiden avulla kaikkea tätä tietoa voidaan hyödyntää järkevästi. Yksi merkittävimmistä lähestymistavoista datan hyödyntämiseen on koneoppiminen. Tämä väitöskirja keskittyy luokitteluun, joka on yksi koneoppimisen osa-alueista. Luokittelulla tarkoitetaan algoritmeja, jotka oppivat datasta sääntöjä, joita voidaan käyttää havaintojen luokkien ennustamiseen.

Tämä työ käsittelee koneoppivaa luokittelua yleisesti ja lisäksi kahta erityistä siihen liittyvää ongelmaa: epätasaisen datan ja tekstin luokittelua. Epätasainen data tarkoittaa sitä, että luokittelutehtävässä yksi tai useampi luokista on huomattavasti harvinaisempi kuin toinen tai toiset. Harvinaisen luokan havainnoilla on yleensä suuri merkitys luokittelutehtävässä. Ongelma on se, että (ilman muokkauksia) monet koneoppimisalgoritmit eivät opi löytämään näitä harvinaisia havaintoja. Tekstin luokittelu puolestaan tarkoittaa luokittelutehtäviä, joissa havainnot ovat luonnollista kieltä.

Tämä väitöskirja sisältää johdatuksen luokitteluun ja tekstidatan analysointiin sekä kolme tieteellistä julkaisua. Ensimmäinen julkaisu soveltaa tekstin luokittelua taloudellisen epävarmuuden mittaamiseen Suomessa hyödyntäen uutisotsikoita. Toinen julkaisu käsittelee epätasaisen datan ja tekstin luokittelua yhtenä kokonaisuutena; esittelemme uuden menetelmän, joka huomioi samanaikaisesti tekstiaineiston ja datan epätasaisuuden luokitteluun tuomia haasteita. Kolmas julkaisu käsittelee – mahdollisesti yllättävän haastavaa – kysymystä siitä, miten eri luokittelijoita tulisi arvioida ja vertailla, kun data on epätasaista.

Preface

Pretty much ever since I first realized that I really like math, doing research was a dream of mine. However, that dream faded away a bit when I started studying math in university and I realized how awfully difficult it can be. Funny enough, that realization never changed—I just kind of figured out that it is a part of the business and that doing research basically starts from that you try to understand something that you don't understand.

First and foremost, I want to thank my wonderful supervisor, Pauliina Ilmonen. I am almost certain that I would not have become a doctor[1] if I had not met Pauliina. Pauliina is a perfect combination of heart-warming kindness and stone-cold brilliance. Working with Pauliina is a joy; you get encouragement and guidance when you need them, you feel the trust and freedom in your work, and sometimes you get that kick to move forward when you are stuck in a loop (something that I definitely required from time to time and really appreciated). I feel great pride saying that Pauliina is not only my supervisor, but also my colleague and a friend.

I am grateful to Professor Thomas Verdebout and Professor Jukka Lempa for taking the time to pre-examine my dissertation. I also thank Professor Thomas Verdebout for agreeing to be the opponent for my doctoral defense. In addition, I gratefully acknowledge the financial support from the Research Council of Finland as well as from the Emil Aaltonen Foundation that have helped me focus on my research during the doctoral studies.

I want to thank Markku Lehmus, with whom I co-authored my first ever scientific article, and who I consider to be my very first mentor in academia. In addition, I have to thank Jetro Anttonen for encouraging me to take on the challenge of doctoral journey, and with whom I have had the pleasure to share countless of interesting conversations over (way too) long lunches.

I cannot thank my wife Elviira enough for all the trust, support, and loving that I have received during our eleven and half years together. I would not have got through the doctoral studies without Elviira by my

[1] Though, at the time I am writing this, I most certainly am not a doctor yet, so fingers crossed...

side. It never ceases to amaze me how she is able to support me not only emotionally but also in the research I have done. I hope I'm able to give back even half of what I have received from you.

I want to thank my family for always supporting me and encouraging me to follow my path. Moreover, I am thankful for all the amazing friends I have made through my time at Aalto, all the wonderful people at the office, as well as my current employer Western Uusimaa Wellbeing Services County for understanding my situation and allowing and helping me to take the time for finishing this thesis concurrently with my job.

Helsinki, October 22, 2025,

Aleksi Avela

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I A. Avela and M. Lehmus. Negative Economic Sentiment Index Based on Finnish News Titles. *Journal of the Finnish Economic Association*, 4(1), 49–63, October 2023.

II A. Avela and P. Ilmonen. Extrapolated Markov Chain Oversampling Method for Imbalanced Text Classification. *Submitted for publication*, available at *arXiv:2509.02332*, September 2025.

III A. Avela. On F_β-score and Cost-Consistency in Evaluation of Imbalanced Classification. In *32nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges (Belgium) and online event, 245–250, i6doc.com, October 2024.

Author's Contribution

Publication I: "Negative Economic Sentiment Index Based on Finnish News Titles"

The idea for the article came from M.L. and A.A. together. A.A. collected and preprocessed the data, implemented the method, and conducted the experiments. M.L. executed the VAR model analyses. The article was written, commented, and revised by M.L. and A.A.

Publication II: "Extrapolated Markov Chain Oversampling Method for Imbalanced Text Classification"

The idea for the article came from A.A. The method was implemented and the experiments conducted by A.A. The experiments were designed by P.I. and A.A. First versions of the article were written by A.A., and P.I. and A.A. commented, polished, and revised the article.

Publication III: "On F_β-score and Cost-Consistency in Evaluation of Imbalanced Classification"

This article was an independent single-authored work conducted by A.A.

List of Figures

List of Tables

Abbreviations

ADASYN Adaptive Synthetic Sampling

AI Artificial intelligence

AUC Area under the (ROC) curve

LDA Latent Dirichlet allocation

LLM Large language model

ML Machine learning

NB Naive Bayes (classifier)

NLP Natural language processing

ROC curve Receiver operating characteristic curve

ROS Random oversampling

RUS Random undersampling

SMOTE Synthetic Minority Over-sampling Technique

SVM Support vector machine (classifier)

TF-IDF Term frequency – inverse document frequency (transformation)

VC dimension Vapnik–Chervonenkis dimension

1. Introduction

Learning is essentially about the leap from past experiences to upcoming decisions. The way both humans and machines learn is fundamentally very similar: try and recognize patterns from known examples and generalize those patterns to adequately accurate rules that can be used for decision-making in the future. Being able to create generalizations from separate examples is also called inductive reasoning or inductive inference—in reference to the fact that learning rarely is a deductive process but is typically based on logical induction. Learning may also incorporate inductive bias, i.e., to consider some generalized rules more plausible than others based on prior knowledge. [76]

Recently, it seems that when people talk about artificial intelligence, or AI, they almost exclusively refer to generative large language models (LLMs). However, AI is a very broad concept that includes basically all programs and algorithms that can make some automated decisions—the list ranges from tic-tac-toe bots to self-driving cars and from email spam filters to chat bots. Nor is AI a new innovation. The earliest advances in modern AI date back to 1940s and 1950s; the foundations for modern deep learning neural networks were laid in a seminal work in 1943 [60] and the famous concept known as the Turing test was introduced by Alan Turing in 1950 [78].

Utilizing the computational power of computers to do tedious, computationally complex or time-consuming tasks makes sense; it frees the time of humans and avoids human errors (though it may bring some other mistakes in exchange as, without explicitly programming it, machines do not tell when they are unsure). Machine learning can be seen as a special case of artificial intelligence where the algorithms learn from examples, i.e., from training data. Machine learning is typically an apt approach if the task is highly complex in a logical sense. For example, even though chess is a complex game, high performing chess engines can be built without machine learning. On the other hand, for example, in tasks related to natural language processing, machine learning approaches typically excel.

Machine learning is typically divided into supervised and unsupervised

learning paradigms. Supervised learning refers to scenarios where the algorithm knows the correct output value or concept for each example in the training data. This thesis focuses on one of the most important concepts of supervised learning: classification. The aim of classification is to learn relations between observations and their respective classes (i.e., concepts or labels) such that also previously unseen observations could be classified as accurately as possible. [76]

On the other hand, unsupervised learning means that the algorithm has only the examples and tries, for instance, to group similar observations together (a process which is called clustering). Unsupervised learning is often also referred to as data-mining, as the aim is typically to find patterns or rules from large data sets that can subsequently be analyzed and rationalized about by humans. Another important category of machine learning is reinforcement learning, where the learning algorithm acts as an agent in an action-feedback environment (in the real-word or in a computer simulation) where it takes turns in making decisions and learning from the feedback it receives. [63]

As said, learning typically cannot be approached with deductive reasoning as the *patterns* and *rules* in real world are highly complex and often even appear to have stochastic elements. In classification, it is virtually impossible (apart from some trivial cases) to find a perfect generalization that would be 100% accurate for all known and all future examples. Thus, it is better to accept that classifiers do make mistakes—the question is about which mistakes to make. In real-life decision-making, every action, be it the correct one or not, has some costs and/or benefits associated with it. While costs and benefits may commonly be understood as referring to monetary losses and gains, they can also be almost anything else—time, respect, or life years. In any case, it is clearly important to try to take into account the associated costs and benefits when making decisions.

The costs (and benefits) are often different for different classes. Moreover, the imbalance in costs is commonly also reflected in class frequencies such that observations belonging to a rare class cost more if misclassified than the ones belonging to a frequent class. Consider, for instance, a classifier which is trying to identify patients with an elevated risk of developing a very rare but potentially life-threatening disease. As the vast majority of patients do not have the risk, a trivial classifier that gives every patient a “no risk” label yields a very high accuracy. However, misclassifying a patient with an elevated risk bears potentially much higher costs than misclassifying a patient with no risk (though, the latter misclassification probably is not cost-free either).

The case presented above is an example of a phenomenon referred to as the problem of imbalanced data (and cost-sensitivity) in classification, and it is the main theme of this thesis. Research of imbalanced data focuses on developing approaches and measures for making optimal decisions when

the class probabilities and misclassification costs are not evenly distributed. [28, 37, 57] As Charles Elkan put it: "The essence of cost-sensitive decision-making is that it can be optimal to act as if one class is true even when some other class is more probable" [28].

In addition, different domains of data may generate their own distinctive challenges in classification—both when the data is balanced and often amplified when it is not. For example, in some classification tasks, e.g., in classification of natural language, the observations are not inherently in numerical form and need to be transformed into numerical values first in order to be digestible for classification algorithms [51]. This feature of text data produces non-trivial challenges both in classification in general as well as when the data is imbalanced [14]. On top of the problem of imbalanced data in general, this thesis also considers the challenges of data imbalance in particular when working with natural language data.

The rest of this thesis is organized as follows. Chapter 2 provides an introduction to statistical learning theory and machine learning classification. Chapter 3 considers the phenomena of imbalanced data and cost-sensitivity and the general challenges related to them. In Chapter 4, the main categories of approaches to imbalanced classification as well as the distinctive challenges of imbalanced text data are discussed. Finally, Chapter 5 summarizes the articles of this thesis, Chapter 6 discusses the significance and future prospects of the work, and Chapter 7 considers certain key terms of the thesis.

2. Classification

Classification is the task of assigning a label (or multiple labels) from a predefined set of classes for a given observation. Each observation consists of a known set of measured variables (also known as features) based on which the classifier tries to predict the correct label(s) for the observation. [63, 76]

The simplest form of classification is the binary case where there are only two classes—often referred to as the positive and the negative class. That is, binary cases are typically of the form of a "yes or no" question, e.g., should this email be labeled as spam, is this credit card transaction fraudulent, or does this patient have a high risk of having a specific disease. On the other hand, in multiclass classification there are more than two (exclusive) classes to choose from, and in multi-label classification each observation can be assigned one, multiple or none of the (non-exclusive) labels. [63, 76]

There are multiple ways to approach classification tasks. These include, for instance, logic-based decision trees and statistical and probabilistic algorithms. This thesis focuses on supervised (statistical and machine learning) classification. The starting point of supervised approaches is to acquire suitable training data, that is, correctly labeled examples[1] which are used for optimizing (i.e., *learning*) a classification rule that can be used for labeling previously unseen instances.

In this chapter, Section 2.1 provides a brief introduction to statistical learning theory, Section 2.2 discusses supervised machine learning classification algorithms, and Section 2.3 considers key aspects of classifier evaluation.

[1] It is not always possible to ensure that every label in the training set is correct—thus, in practice, learning applications may have to be able to handle partly incorrect or noisy training data as well.

2.1 Foundations of Statistical Learning

In the context of classification, the goal of learning is to use a labeled training sample for finding an accurate mapping from the input space to the output space. The input space (also often called the feature space in machine learning literature) consists of some measured variables of the observations and the output space includes the possible classes (or labels) for the observations. A training sample is a set $T = (X_i, Y_i)_{i=1}^n$ of examples $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. In classification, the output space is discrete (and finite), whereas regression deals with learning tasks with continuous output spaces. [52, 63, 76]

2.1.1 Classifier Learning

In some cases, it is adequate to assume that there is an unknown but deterministic underlying mapping f from input space to output space, and only the observations are sampled from a distribution defined on $\mathcal{X}$ but the labels are deterministic for every observation. However, generally the case is that there is some uncertainty in the labels as well (for instance, due to the fact that the selected set of features cannot convey all relevant information for mapping from the input space to the output space). Thus, the more common case is to regard the classification task as stochastic and to assume that the examples are drawn (i.i.d.) from a joint distribution ρ defined on $\mathcal{X} \times \mathcal{Y}$. [63]

A classifier h is a mapping $h : \mathcal{X} \mapsto \mathcal{Y}$ from the input space to the output space, and the aim is to be able to use the classifier for any observations drawn from the same distribution ρ as the training sample. The two main concerns of a classification algorithm are the ability to find a suitable classification rule in the training (in-sample) data, and the ability to fit a classification rule that also generalizes to other (out-of-sample) data drawn from the same distribution[2]. [63, 76]

Finding the best, or even an adequate, decision rule for the training set is rarely trivial, especially as many real-life data sets are not (linearly) separable. Yet, as the modern learning and optimization algorithms (and computers) are so powerful, typically the more challenging task is to regularize the classification rule, that is, to make the classifier generalize well to out-of-sample data and not to *overfit* the training data.

The quality of a classifier h is evaluated based on a loss function (often also called a cost function) $l(h(X), Y)$. Combining the distribution ρ and the loss function gives us the definition of risk:

$$R(h) := E_\rho[l(h(X), Y)]. \tag{2.1}$$

[2] In some learning tasks, the joint distribution ρ may change over time—either smoothly or at one (unknown) point of time. This specific problem of learning is called concept drift (see, e.g., [55]).

However, as the distribution ρ is practically never known in real-life tasks, empirical risk $\hat{R}(h) := \frac{1}{n}\sum_{i=1}^{n} l(h(X_i), Y_i)$ is used instead. The risk of the population is also referred to as the generalization error, which is commonly estimated as the empirical risk of out-of-sample data, i.e., as the average loss in the so-called test set (data from distribution ρ which was not used in learning the classification rule). One of the most used loss functions is the 0-1 loss: $l_{0\text{-}1}(h(X), Y) = \mathbb{1}[h(X) \neq Y]$. [63, 76]

Let $\mathcal{F}$ denote the set of measurable functions $\mathcal{X} \mapsto \mathcal{Y}$. The (theoretical) optimal Bayes model is then defined as $f^* := \arg\min_{f \in \mathcal{F}} R(f)$. In practice, the learning approach is commonly to select a subset $\mathcal{H} \subset \mathcal{F}$ as a set of possible models (known as the hypothesis space) and try to find the classifier as $h^* = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$. The difference between the risk of a theoretical Bayes model f^* and the risk of a model that is achievable in practice h^* is called the excess risk, and it can be decomposed as:

$$R(h^*) - R(f^*) = \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}}, \tag{2.2}$$

where $\mathcal{E}_{\text{app}}$ is the approximation error (also known as bias in machine learning literature) and $\mathcal{E}_{\text{est}}$ is the estimation error (also known as variance). The approximation error is caused by the choice of $\mathcal{H}$ and the estimation error by the selected classifier and the training sample T. [63, 76]

In practice, the set of possible models $\mathcal{H}$ is typically implicitly induced based on the functional form of the classifier h (and its hyperparameters); for instance, whether the chosen model approaches the classification task with a linear or a non-linear decision rule. An alternative decomposition for Equation 2.2 takes into account also an error caused by the optimization approach such that

$$R(h^*) - R(f^*) = \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}, \tag{2.3}$$

where the optimization error $\mathcal{E}_{\text{opt}}$ is caused by that, in practice, the model is fitted as $h^* = \arg\widehat{\min}_{h \in \mathcal{H}} \hat{R}(h)$, where $\widehat{\min}$ is some practical optimization algorithm used for learning the classification rule [12]. In this decomposition, the errors are

$$\begin{aligned}
\mathcal{E}_{\text{app}} &= R(h^{\text{app}}) - R(f^*), \\
\mathcal{E}_{\text{est}} &= R(h^{\text{est}}) - R(h^{\text{app}}), \\
\mathcal{E}_{\text{opt}} &= R(h^*) - R(h^{\text{est}}),
\end{aligned}$$

where $h^{\text{app}} = \arg\min_{h \in \mathcal{H}} R(h)$, and $h^{\text{est}} = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$.

2.1.2 Balancing the Errors in Learning

Commonly, there are trade-offs between the errors listed in the previous section. The approximation-estimation trade-off (i.e., bias-variance trade-off) is generally relevant regardless of the scale of the learning task [12].

A rule of thumb in machine learning is that, particularly when dealing with limited training data, inducing some bias (approximation error) in the model by restraining the size of $\mathcal{H}$ can reduce the effective variance (estimation error) so much that the total error is reduced as well. That is, on top of empirical risk minimization, learning algorithms often also include inductive bias in some form. The reason being that not restricting the hypothesis space and only focusing on minimizing the in-sample empirical risk can lead to a too specific decision rule (i.e., overfitting) that works well in the training set but generalizes poorly to out-of-sample data. [76]

Technically speaking, though, practically all learning approaches consist of real-valued parameters, thus making the size of the effective hypothesis space infinite. However, there are other means for measuring the complexity of $\mathcal{H}$, such as Vapnik–Chervonenkis (VC) dimension [81][3]. For a classification task with a p-variate input space $\mathcal{X}$ and a binary output space $\mathcal{Y}$, the VC dimension of hypothesis space $\mathcal{H}$ is defined as the size of the largest set of any points in $\mathbb{R}^p$ that a classifier $h \in \mathcal{H}$ is able to classify perfectly for an arbitrary labeling of those points. That is, the VC dimension of a binary classifier h for a learning task $\mathcal{X} \mapsto \mathcal{Y}$ can be regarded as a measure of the complexity of $\mathcal{H}$. VC dimension is defined only for binary classification tasks, whereas Natarajan dimension [65] is a complexity measure that generalizes the concept to multiclass tasks (i.e., tasks where $|\mathcal{Y}| > 2$).

The issue, however, is that the VC dimension of very complex models is infinite. There exists also other approaches, such as Rademacher complexity (see, e.g., [76]), which may be more useful for complex classification models and tasks. Unlike VC dimension, Rademacher complexity depends also on the given training sample and measures the complexity of $\mathcal{H}$ based on the average empirical error of best possible classifiers $h \in \mathcal{H}$ for training sets where the labels have been drawn randomly from $\mathcal{Y}$. That is, it measures complexity of $\mathcal{H}$ as its ability to fit randomly drawn labels—the smaller the average error, the higher the complexity of $\mathcal{H}$.

The bias-variance trade-off shown in Equation 2.2 can be balanced by choosing a model with a simpler functional form, thus reducing the size of $\mathcal{H}$ and increasing the approximation error. A simpler model overfits the training data less than a more complex model, and therefore, the estimation error often decreases more than the approximation error increases. On the other hand, overfitting is less of a problem when the training sample is very large—however, large-scale learning tasks are subject to the effect of optimization error (Equation 2.3) in non-trivial ways [12]. Another way to balance the bias-variance trade-off is to use regularization techniques. These include, e.g., adding a penalty term depending on the magnitudes of the model parameters (the sum of l^2 or l^1 norms are popular choices) to

[3]English translation of the original Russian paper (1968).

the objective function that is minimized when fitting the classifier. [63, 76]

In practice, probably the most used approach to restrain the estimation error is k-fold cross-validation, which is a general approach that can be used when optimizing the model hyperparameters. Hyperparameters (or free parameters) refer to the parameters of a model that are not optimized in learning but are set prior to fitting the classifier. They may, for example, have an effect on the functional form and complexity of the classification rule or on the weights assigned to different classes. [63]

The idea of k-fold cross-validation is to divide the training set randomly into k (roughly) equal size sub-samples (called folds). Each fold is used at a time as a hold-out set and the sample consisting of the remaining $k-1$ folds is used for fitting the classification rule. In each iteration, the resulting classifier is evaluated using the hold-out set, and after all k iterations have passed, the evaluations are averaged. Let $F^j = (X^j, Y^j) \in T$ denote the j:th fold of the training sample $T = (X_i, Y_i)_{i=1}^n$. The cross-validated empirical risk $\hat{R}_{\text{cv}}(h)$ is then:

$$\hat{R}_{\text{cv}}(h) = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{1}{n_j} \sum_{i=1}^{n_j} l\big(h_{T \setminus F^j}(X_i^j), Y_i^j\big) \right),$$

where $h_{T \setminus F^j}$ means a classifier learned from the sample $T \setminus F^j$ and n_j is the number of observations in the fold j. [63]

The intuition behind k-fold cross-validation is that, as the aim is to achieve good generalization for out-of-sample data, in each iteration, the classifier is evaluated using a hold-out set, while still being able to make use of the whole training sample by iterating over the folds. On the other hand, if the training sample is very large and/or the applied classification algorithm is highly complex, the computational time of k-fold cross-validation may also be high.

2.2 Supervised Machine Learning

Machine learning can be broadly divided into three main categories: (i) supervised, (ii) unsupervised, and (iii) reinforcement learning. Supervised learning deals with tasks with prelabeled training data, whereas unsupervised learning (also known as data mining) aims at learning interesting properties and patterns (e.g., sub-clusters) from unlabeled data. Reinforcement learning is an approach for dynamic input-output environments, where the algorithm acts as an agent that simultaneously aims at both learning its environment and making optimal decisions. Other learning scenarios include, for instance, online learning (where, instead of using a large training set before operation, the algorithm receives training data while operating), and active learning (where the algorithm has, e.g., an option to not classify difficult examples itself but ask for an expert opinion

and learn from it). [52, 63, 76]

Classification algorithms that are considered to fall under the branch of machine learning typically have (at least some of) the following properties: (i) they make few or no distributional assumptions about the data generating process, (ii) they do not have closed-form solutions, and (iii) they are based on an iterative (and heuristic) search of the hypothesis space with, e.g., stochastic gradient descent.

Some standard machine learning algorithms are inherently designed for binary classification, and in order to apply them for multiclass data some modifications are needed [7]. One approach is to transform multiclass classification into multiple binary tasks with, e.g., one-vs-rest or one-vs-one schemes (see, e.g., [31]). For a multiclass task with k classes: $[y_1, y_2, ..., y_k]$, one-vs-rest approach generates k different binary tasks where y_i is the positive class and the remaining classes together compose the negative class. On the other hand, for the same task, one-vs-one approach produces $\frac{k(k-1)}{2}$ tasks where y_i is the positive class and y_j is the negative class, where $j \neq i$. The labels produced with these approaches (in an out-of-sample set) are aggregated into final classifications, for instance, based on voting. [7]

As machine learning algorithms typically rely heavily on the labeled training sample (often accompanied with some regularization) to create classifiers with good generalization, the quality of data is of crucial importance for machine learning approaches. Many classification algorithms are unable to handle missing values. If there are only few examples (or features) that miss values in the training set, a common approach is just to remove observation rows (or feature columns) with missing values in the data preprocessing phase. [92]

If the missing values are distributed over multiple rows and/or columns, deleting observations or features with missing values bears a high risk of losing important information. Thus, there have been a lot of research focusing on approaches to impute the missing values (for example, based on the feature mean across the data or within the given class) in order to avoid removing observations or features from the sample (see, e.g., [3] and [29]). Another important aspect of data preprocessing is feature selection and analysis of feature relevance (see, e.g., [34]). Feature selection can also be regarded as a form of regularization as it effectively decreases the model complexity by discarding features with the lowest relevance [34].

The set of different classification approaches is extremely broad and greatly varies in the algorithm complexity—from the earliest ideas, such as (single-layer) perceptrons [72] and k-nearest neighbors [18], to multilayer perceptrons and ensemble methods, such as Adaptive Boosting [33] and random forests [42], and to the modern deep learning neural networks. The following two sections provide an introduction to two widely used classification algorithms, naive Bayes classifier and support vector machines, which both have also been applied in the articles of this thesis.

2.2.1 Naive Bayes Classifier

Naive Bayes (NB) classifier (see, e.g., [7, 23, 51, 76]) is a probabilistic classification approach that is based on an assumption that the features of each observation are mutually independent. Based on this assumption, the likelihood of an observation can be computed as the product of the conditional probabilities of its feature values independently. An instance x_j is classified to class i that maximizes the posterior probability of the class. For computational reasons, the classification is typically based on log-probabilities:

$$\arg\max_i \big(\log f(i) + \log f_i(x_j) \big),$$

where $f(i)$ is the prior probability of class i and $f_i(x_j)$ is the likelihood of observation x_j in class i.

Let observation $x_j = (x_j^1, ..., x_j^p)$ be a p-dimensional vector drawn from a multinomial distribution. Furthermore, let each of the considered classes be represented with a distinct multinomial distribution characterized by a parameter vector $\theta_i = (\theta_i^1, ..., \theta_i^p)$. This approach is known as multinomial NB, and it is often used, e.g., in text classification with a bag-of-words modeling approach (see, Section 4.4.1). Based on the feature independence assumption, the log-likelihood in multinomial NB is given by

$$\log f_i(x_j) = \sum_{k=1}^{p} \big(x_j^k \log \theta_i^k \big). \tag{2.4}$$

Let $(x_j, y_j) \in T$ be a set of n training observations and their respective labels. In multinomial NB, the parameter vector estimates $\hat{\theta}_i$ are commonly computed as smoothed maximum likelihood estimates:

$$\theta_i^k = \frac{N_i^k + \alpha}{N_i + \alpha n},$$

where $N_i^k = \sum_{x_j \in T | y_j = i} x_j^k$ is the number of occurrences of feature k in class i, and $N_i = \sum_{k=1}^{p} N_i^k$ is the total number of occurrences of all features in class i. The pseudo count $\alpha > 0$ serves as a smoothing prior to prevent zero estimates. This technique is commonly called Laplace smoothing—though, some authors reserve this term only for the case $\alpha = 1$. [51]

For instance in text classification, on top of smoothing certain other transformations—such as TF-IDF weighting (see, [74])—are typically applied when estimating the parameters, see, e.g., [71]. Text classification is a great example of the ability of NB to produce relatively accurate classifications even though the features (i.e., words in a text) are clearly dependent [23]. Another advantages of NB classifier include its computational lightness and high interpretability of its parameters. On top of the multinomial model, other versions of naive Bayes include, for instance, Bernoulli NB with binary variables and Gaussian NB with normal variables.

2.2.2 Support Vector Machines

Support vector machine (SVM) [11, 17] is one of the most used machine learning classification algorithms. SVMs, and especially the non-linear kernel extensions, are very flexible and can achieve state-of-the-art level performance in a variety of tasks. The idea of SVM is to fit a maximum-margin hyperplane in the p-dimensional feature space to separate the two classes with labels -1 and 1.

In the simplest case, where the two classes are linearly separable in the training sample, the original version of the algorithm, hard-margin SVM, can be used. Hard-margin SVM finds the two parallel hyperplanes that fully separate the classes while maximizing the margin between the hyperplanes. These hyperplanes are represented with the set of points x that satisfy $w^\top x + b = \pm 1$, where the normal vector w and intercept b are the fitted parameters. The actual decision boundary used in classification, $w^\top x + b = 0$, lies between the margin hyperplanes.

The distance between the margin hyperplanes is $\frac{2}{\|w\|}$. Maximizing this distance is (computationally conveniently) equivalent to minimizing $\frac{1}{2}\|w\|^2$, leading to the optimization problem:

$$\begin{aligned} \arg\min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_j(w^\top x_j + b) \geq 1,\ j = 1, ..., n, \end{aligned}$$

where $(x_j, y_j) \in T$ is the training set of observations and their binary labels and n is the size of T. In hard-margin SVM, the observations that lie on either margin (i.e., observations x_j for which $y_j(w^\top x_j + b) = 1$) are called the *support vectors*—giving the classifier its name.

Real-life data, however, rarely is linearly separable. The soft-margin version of SVM [17] (which is the algorithm people are typically referring to when talking about SVM) handles non-separable data by introducing hinge loss slack variables $\xi_j = \max\left(0, 1 - y_j(w^\top x_j + b)\right)$ for observations that fall on the wrong side of the margin hyperplanes. With the slack variables, the optimization problem becomes:

$$\begin{aligned} \arg\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{j=1}^{n} \xi_j \\ \text{s.t.} \quad & y_j(w^\top x_j + b) \geq 1 - \xi_j,\ j = 1, ..., n, \\ & \xi_j \geq 0,\ j = 1, ..., n, \end{aligned}$$

where $C > 0$ is a hyperparameter controlling the trade-off between margin width maximization and loss term minimization.

Although SVM is a linear classifier, it can be applied to non-linear classification tasks as well by using the *kernel trick* [11]. The dual of the

SVM optimization problem can be written such that is depends only on dot products between the observations. The idea of the kernel trick is to replace the dot product with a kernel function that defines an inner product in a higher dimensional space. A classification problem can be solved with a linear classifier in the transformed space by utilizing the powerful (hard-margin or soft-margin) SVM algorithm. The resulting classifier, however, accounts for a non-linear classifier in the original feature space. Some widely used kernels include, for example, polynomial, sigmoid, and radial basis function (RBF) kernels. For more on kernel functions and optimization of SVM, see, for instance, [63] and [76].

2.3 Classifier Evaluation

In real-life classification tasks, empirical risk is often accompanied with some other statistics in evaluation in order to give a broader view of the classification performance. On the other hand, the exact loss function is not always known, but simply using a generic 0-1 loss is problematic as the considered classes may have different sizes and importance. Accuracy (which is arguably the most used evaluation measure) is a direct counterpart to 0-1 loss, as accuracy is the number of correctly classified observations divided by the total number of observations, which is equal to one minus empirical risk with 0-1 loss. [16, 28, 57, 85]

In a binary case, where the output values consist of a positive and a negative class, the performance of a classifier on any (labeled) data sample can be fully summarized in a two-by-two confusion matrix, as shown in Table 2.1. A binary confusion matrix includes four values: tp (number of true positives), tn (number of true negatives), fp (number of false positives), and fn (number of false negatives). However, as comparing multiple classifiers based on these four values (let alone, more than four values, if there are more than two classes) simultaneously is typically challenging, confusion matrices are commonly compressed into univariate evaluation statistics. [16, 57]

Table 2.1. Binary confusion matrix.

Actual:		**Positive**	**Negative**
Predicted:	**Positive**	tp	fp
	Negative	fn	tn

Simple univariate statistics include, e.g., true positive rate (tpr, also known as recall or sensitivity) $\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$, true negative rate (tnr, also known as specificity) $\frac{\mathrm{tn}}{\mathrm{tn}+\mathrm{fp}}$, and precision (also known as positive predictive value) $\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$. Tpr and tnr measure the share of correctly classified positive and

negative observations, respectively, and precision measures the share of actual positives out of all positive predictions. [57, 76, 85]

On their own, however, these simple univariate statistics are too simplified, as perfect tpr (or tnr) can be obtained by classifying every observation positive (or negative). Perfect precision is not trivially achievable, but typically the case is that the fewer observations are classified positive the higher the value of precision. To remedy these problems, the most popular classification evaluation statistics are based on combining some of the simple statistics listed above. [57] These include, for instance, balanced accuracy, which is the arithmetic mean of tpr and tnr, and F_1-score, which is the harmonic mean of precision and recall (tpr).

F_1-score is a special case of a more generic evaluation statistic, F_β-score:

$$F_\beta = (1+\beta^2)\frac{\text{precision} \times \text{recall}}{(\beta^2 \times \text{precision}) + \text{recall}}. \tag{2.5}$$

F_β-score is often recommended to be used especially when dealing with imbalanced data (imbalanced data is further discussed in Chapter 3), but recently, some researchers have raised questions on whether F_β-score in fact is a suitable statistic for evaluation of imbalanced classification, see, e.g., [16, 35, 69]. Publication III of this thesis provides some new insights on the drawbacks of applying F_β-score for evaluation of imbalanced classification and compares it to another evaluation measure, informedness.

Informedness is defined as $\text{tpr} + \text{tnr} - 1$, that is, it puts equal emphasis on tpr and tnr (see, e.g., [69]). Another widely used evaluation measure in imbalanced classification is balanced accuracy, defined as $\frac{\text{tpr}+\text{tnr}}{2}$ (see, e.g., [85]). Balanced accuracy is essentially the same measure as informedness—the only difference being that informedness ranges from -1 to 1 whereas balanced accuracy ranges from 0 to 1. As discussed in Publication III, unlike F_β-score, informedness and balanced accuracy behave linearly compared to total cost when the misclassification costs are observation-independent and linear. Total classification cost is further discussed in Section 3.2.

The evaluation statistics discussed above are designed exclusively for the evaluation of some specific classification rule. However, there exists also "global" measures that aim to present a broader assessment of the capabilities of a classifier. Many classification algorithms offer an option for moving the decision boundary via classification threshold adjustment. For example, as discussed earlier, the default behavior of SVM is to classify observation x_j based on whether $w^\top x_j + b$ is greater or less than zero. However, this classification threshold can be freely adjusted higher or lower as well, e.g., depending on the misclassification costs of the given classification task.

Arguably the most used global evaluation measure is the receiver operating characteristic (ROC) curve. The ROC curve plots tpr of a classifier

against its false positive rate (i.e., $1 - \text{tnr}$) by increasing the classification threshold based on some step interval. The range of evaluated thresholds is set such that the end-points of a ROC curve are at $(0, 0)$ and $(1, 1)$. The closer the curve gets to the top-left corner the better, where the top-left point accounts for a perfect classification, i.e., $\text{tpr} = 1$ and $\text{fpr} = 0$. [63]

Precision-recall curve is another widely used global measure. As the name suggests, it plots precision of the classifier against its recall (tpr) by altering the classification threshold, where the optimal point is the top-right corner, i.e., when both recall and precision are equal to one. ROC and precision-recall curves may also be compressed into univariate measures, e.g., by computing the area under the ROC curve (AUC) value or the average precision (AP) value. [20]

Examples of ROC and precision-recall curves and respective AUC and AP values are illustrated in Figure 2.1. These examples consider the performance of two classifiers, SVM with a default linear kernel and SVM with a non-linear RBF kernel, on a simple simulated data set. In both cases, the non-linear SVM classifier is said to dominate the linear SVM classifier as, in every point, its curve is higher than (or equal to) the other curve [20]. Note that better AUC or AP values do not guarantee that one classifier dominates the other, but on the other way around, dominant classifier always has a higher value of AUC or PR—as is the case in this example, too.

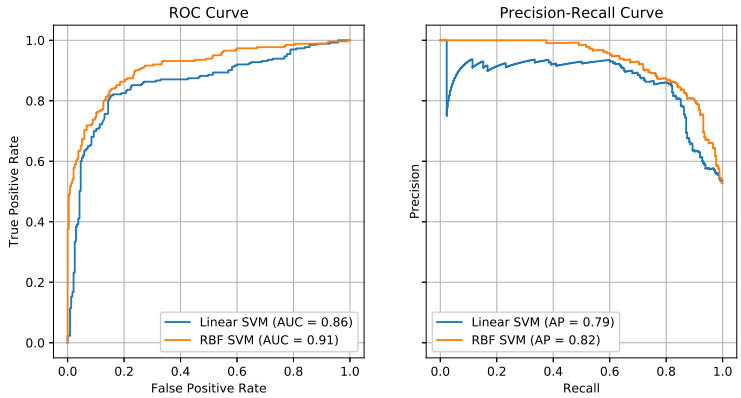

Figure 2.1. ROC and precision-recall curves on simulated data.

One of the challenges of classification evaluation is that it is sometimes difficult to keep different comparisons comparable as there are an abundance of evaluation measures to choose from. In research this means that there has to be some established conventional measures that make research evaluations comparable and transparent. On the other hand, in practical applications, the practitioners need to know which measures are the most informative for their case.

3. Imbalanced Data and Cost-Sensitivity

In classification problems, data is said to be imbalanced if the number of observations varies significantly across the considered classes. There does not exist any specific threshold for data to be imbalanced, and technically, any data set that features an uneven class distribution can be called imbalanced. However, since lower imbalance ratios are not known to cause similar issues in classification as higher ratios, the imbalance ratios considered in the literature are typically at least of the order of 1:100 [37]. On the other hand, there does not exist an upper-limit for class imbalance, and, in theory, imbalance ratios can be arbitrarily large.

In a binary imbalanced classification task, the smaller class is commonly called a minority class, and correspondingly, the larger class is called a majority class. Furthermore, the minority class is often labeled as the positive class, since the minority instances typically constitute the set of interesting observations that are wished to be detected among the majority (negative) observations. [13, 70, 85, 87]

Research of imbalanced data often focuses on the binary case, as classification of multiclass data is usually conducted by dividing the problem into a set of binary tasks with either one-vs-one or one-vs-rest scheme. The higher the number of classes, the higher the degree of imbalance typically is in the data. [7, 45] Nevertheless, when dealing with a classification task that includes a high number of classes (either in multiclass or multi-label classification) the challenges caused by different types of class distributions may also need to be considered, see, e.g., [40].

One of the most common issues with imbalanced data is that standard classifiers have a tendency to fit a classification rule with a trivially high (in-sample) accuracy while favoring the majority class and neglecting the minority class [1, 37, 47, 49, 58, 89]. Many standard classification algorithms are designed, either directly or indirectly, to maximize accuracy—thus assuming that all misclassifications are equally costly [89, 90]. In reality, however, the minority instances are typically of high importance and the cost of misclassifying a minority observation is much higher than the cost of misclassifying a majority observation. Thus, in most cases, accu-

racy is not a suitable measure for decision rule optimization and evaluation in imbalanced classification [57, 90].

The challenges of imbalanced classification can be broadly divided into two main sources: (i) the issues related to the characteristics of imbalanced data that hinder generalization (of minority class), and (ii) the inability of many learning algorithms to deal with uneven class prior and cost distributions. The challenges related to classification of imbalanced data and to cost imbalances are discussed in Sections 3.1 and 3.2, respectively.

3.1 Characteristics of Imbalanced Data

It should be noted that, strictly speaking, the concept of imbalance merely refers to the fact that the given (training) data set is imbalanced—though, typically the interest is on the underlying (unknown) distribution [87]. However, usually it is adequate to assume that the sample class distribution is an accurate estimate of the population distribution, and that any future sample from the same population to which the classifier is applied should feature a similar degree of imbalance as the training set. Yet, this is not always the case, and the underlying distribution (and the costs of misclassifications) may change in time or from place to place. [70]

Sample imbalance that can be assumed to reflect the imbalance in the underlying distribution is called intrinsic imbalance as it is assumed to be a true property of the population and not an artifact related to the considered data set. On the other hand, imbalance that is not caused (only) by the underlying class distribution but (also), for instance, by some bias in data gathering process, is referred to as extrinsic imbalance. Identifying and modeling extrinsic imbalance (and its sources) can be extremely challenging in real life. [37]

Imbalances can also be categorized into relative and absolute rarity [37, 87]. Relative rarity is the more typical case, where the imbalance in data is caused by that minority observations are *relatively* rarer than majority observations. That is, the minority class is smaller than the majority class, but if the training set is large enough, there may still be a plenty of information available about the minority class. Of course, it is rarely possible to just arbitrarily increase the training set size—making relative rarity problematic for classification. On the other hand, in case of absolute rarity, the size of the training set has little effect on the fact that information about the minority class is very limited. The most severe cases of absolute rarity are considered in the field of novelty detection, where there is not a single training observation representing the "minority class" (see, e.g., [67]).

The challenges of imbalanced classification are not limited to just between-class imbalance, but imbalanced data often comes with a set of other issues

that make learning more difficult [57]. For example, imbalanced data often also includes within-class imbalances, that is, the classes themselves consist of multiple (imbalanced) sub-clusters [48]. The issue with having an internally imbalanced minority class is that, as there are only a limited number of minority training observations, parts of splintered minority class may be treated as noise or outliers, and consequently, generalizing the minority class becomes highly difficult for learning algorithms.

Moreover, related to the issue of within-class imbalance, imbalanced data is often associated with the problem of small disjuncts [44, 49, 88]. Many algorithms fit a classifier based on multiple disjuncts in the data space [49]. However, correctly classifying minority observations often requires finding "small disjuncts", as detecting minority instances—which, by the nature of imbalanced data, are rare—requires fitting very specific classification rules. The number of training observations that a disjunct correctly classifies is referred to as its coverage, and in small disjuncts that coverage is low. [49]

Due to different regularization techniques that the classifiers apply, small disjuncts that in fact account for the minority class may get dropped from the classification rule [44, 49, 57, 88]. Moreover, in real-life, data sets often have high dimensionality which makes detecting low coverage areas in data even more difficult [49]. At the same time, however, regularization is usually necessary in order to avoid overfitting, as (at least in balanced data sets) small disjuncts are often caused by noise and removing them is justified [87].

Other collateral difficulties of imbalanced data include, for instance, the issue of generalizing the minority class as there is a lack of information in the training data [57, 87]. Moreover, imbalanced data is known to be prone to noise in data [1, 57, 75]. Together, noise and lack of information in training data also introduce the issue of data set shift, i.e., noisy data and overfitting may cause poor out-of-sample performance [57]. The difficulties of imbalanced classification are commonly amplified by data complexity, e.g., high dimensionality and class overlapping [49, 57]. In fact, class overlapping can be regarded as one of the main challenges of classification in general. Namely, if the classes are fully (linearly) separable, no learning algorithm should have any problems fitting a perfect classification rule—regardless of whether the data is imbalanced or not. [5, 21, 57]

The final issue of imbalanced classification that should be discussed is the use of improper evaluation measures. As mentioned earlier, accuracy—on which many learning algorithms are inherently based—is not a suitable measure for imbalanced classification as minority classes have only little impact on it [87]. In a binary case, the measures of interest typically are recall (true positive rate), true negative rate, and precision, as discussed in Section 2.3. However, these measures are not able to consider the big picture in classification; for instance, 100% recall can be achieved trivially by classifying every observation to positive class. Thus, many conventional

evaluation measures are defined as combinations of, for example, the three measures listed above. Another measure for evaluation is total cost, which is discussed in the following section.

3.2 Cost Imbalances

A sometimes overlooked property of imbalanced classification is that the class imbalance itself is not that significant of an issue if the misclassification costs are similar for all classes. Sure, imbalance can cause poor generalization of the minority class, but if the costs of a false negative and a false positive are equal, it hardly makes any difference. However, the cost of a false negative classification typically far surpasses the cost of a false positive, as the minority class commonly consists of those interesting or important instances that we want to or need to find [13, 70, 87].

Theoretically, each observation is associated with a cost $C(i, j, x)$ of assigning observation x to class i when the true class is j [90]. However, there is often some uncertainty related to the costs and they are rarely explicitly known in practice [80]. It is important to notice that, despite the term, not all costs are money—in fact, in many real-life applications (some) costs are something else, e.g., time or quality of life [46, 80, 92]. Moreover, even if costs could be measured as money, it can be that they will only be realized as monetary losses or gains over a long time horizon, making the cost estimation highly challenging.

A simple cost structure is such that the costs are independent of the observations and depend only on the true and predicted classes, that is, the cost $C(i, j, x)$ can be written as $C(i, j)$. Moreover, in the simple case, the costs are typically linear, i.e., for instance, twice as many false positives costs twice as much. In a binary case, the simple cost structure is given by four constant cost values: cost of true positive, C_{TP}, cost of false positive, C_{FP}, cost of false negative, C_{FN}, and cost of true negative, C_{TN}. [22, 28, 84].

Similarly to classification confusion matrices (in a binary case) discussed in Section 2.3, the costs can be summarized in a cost matrix where the entries represent the costs of all possible classifications and misclassifications as shown in Table 3.1. The costs can be positive, zero, or negative (i.e., gains). [28, 57, 84, 90]

A *reasonable* cost system can be defined such that $C_{\text{TP}} < C_{\text{FN}}$ and $C_{\text{TN}} < C_{\text{FP}}$, i.e., the cost of classifying an observation correctly is always smaller than the cost of misclassifying it [28]. Moreover, any reasonable cost system can be transformed into

$$C'_{\text{TP}} = \frac{C_{\text{TP}} - C_{\text{TN}}}{C_{\text{FP}} - C_{\text{TN}}}, \qquad C'_{\text{FP}} = 1, \qquad C'_{\text{FN}} = \frac{C_{\text{FN}} - C_{\text{TN}}}{C_{\text{FP}} - C_{\text{TN}}}, \qquad C'_{\text{TN}} = 0, \tag{3.1}$$

while preserving monotonicity with respect to total cost [28].

As the classes are very rarely fully separable, the main issue concerns

Table 3.1. Binary cost matrix.

Actual:		**Positive**	**Negative**
Predicted:	**Positive**	C_{TP}	C_{FP}
	Negative	C_{FN}	C_{TN}

those observations that fall into the overlapping region. The challenge typically lies in the trade-off between false positives and false negatives. In case of a fixed data sample, total classification cost is arguably the most informative evaluation measure—assuming that the cost matrix is explicitly known [89]. Given the numbers of instances in the confusion matrix, tp, fn, fp, and tn, total cost is

$$C = \text{tp}C_{\text{TP}} + \text{fn}C_{\text{FN}} + \text{fp}C_{\text{FP}} + \text{tn}C_{\text{TN}}.$$

That is, the loss function is given by the cost matrix, and total classification cost of the classifier h is $n\hat{R}(h)$, where n is the number of observations in the sample and $\hat{R}(h)$ is the empirical risk in the sample. The transformed form of total cost given by the cost transformations in Equation 3.1 is

$$C' = \text{tp}C'_{\text{TP}} + \text{fn}C'_{\text{FN}} + \text{fp}.$$

Another condition for a reasonable cost system is that all costs should be measured from the same baseline—for instance, opportunity costs (i.e., missed gains) do not follow the same baseline as factual costs [28]. However, assuming a fixed data sample for all the evaluated classifiers, we can set the perfect classification as the baseline, and reduce the cost structure to an even simpler form of $c \times \text{fn} + \text{fp}$, where $c = \frac{C_{\text{FN}} - C_{\text{TP}}}{C_{\text{FP}} - C_{\text{TN}}}$.

In Publication III, it is shown that the transformed total cost can in fact be replaced with this simpler scaled cost in evaluation as long as the data sample is fixed for all the evaluated classifiers. The simple scaled cost we defined is the difference between the transformed total cost of the considered classification and the transformed total cost of a hypothetical perfect ($\text{tp} = \text{ap}$ and $\text{tn} = \text{an}$) classification:

$$C^* = C'_{\text{TP}}\text{ap} + C'_{\text{TN}}\text{an} = C'_{\text{TP}}\text{ap} = C'_{\text{TP}}(\text{tp} + \text{fn}).$$

The difference for the given sample is

$$\begin{aligned}
&C'_{\text{FN}}\text{fn} + C'_{\text{TP}}\text{tp} + C'_{\text{FP}}\text{fp} + C'_{\text{TN}}\text{tn} - C^* \\
=&C'_{\text{FN}}\text{fn} + C'_{\text{TP}}\text{tp} + \text{fp} - C'_{\text{TP}}(\text{tp} + \text{fn}) \\
=&(C'_{\text{FN}} - C'_{\text{TP}})\text{fn} + \text{fp} = \left(\tfrac{C_{\text{FN}} - C_{\text{TN}}}{C_{\text{FP}} - C_{\text{TN}}} - \tfrac{C_{\text{TP}} - C_{\text{TN}}}{C_{\text{FP}} - C_{\text{TN}}}\right)\text{fn} + \text{fp} = \tfrac{C_{\text{FN}} - C_{\text{TP}}}{C_{\text{FP}} - C_{\text{TN}}}\text{fn} + \text{fp} \\
=&c \times \text{fn} + \text{fp},
\end{aligned}$$

which is a strictly increasing function of total cost. That is, given a fixed sample, replacing the actual or transformed cost matrix with this simpler form does not affect evaluation.

Although total cost is the most suitable evaluation measure for a fixed sample when all the classification costs are explicitly known, it is important to remember that—just as with any other evaluation measure—sample total cost is merely an estimate of the expected value of total cost. Moreover, the classification costs are rarely fully known in practice, and they can also be observation-dependent, non-linear, or change in time or from place to place [30, 70, 80, 90]. Estimation of costs in imbalanced classification is further discussed in Section 4.2.

Costs of imbalanced learning and decision making are not limited to just classification costs. In practice, there are multiple steps involved in a decision making process—from data gathering and processing to algorithm training to actual decisions and model tuning. For example, in certain domains, e.g., in medical diagnosis, different tests (i.e., data features) may have imbalanced costs as well. That is, if different classifications and features both have unevenly distributed costs, the learning and decision making process also involves a question of which features should be included in order to produce the possible predictions while minimizing the cost of features. [46, 66, 80, 92]

If both the misclassification and feature costs are explicitly known, they both account for the total (expected) cost of classification, and the question of which features should be involved can be answered explicitly as a single cost optimization problem [79, 80]. There are other possible sources of costs involved in classification as well. These include, for instance, costs of training examples, computational costs, and human-computer interaction costs [46, 80].

4. Approaches to Imbalanced Classification

The approaches to imbalanced classification can be divided into three main categories: data-level methods (i.e., sampling), model-level modifications (i.e., algorithmic approaches), and the theoretical framework of Bayes-optimal cost-sensitive learning. [24, 31, 37, 57] There also exists approaches that combine two or all of the general categories listed above. None of these approaches outperform the others in every application, and approaching imbalanced classification always requires considering the given task, its objectives, and any data-specific challenges.

This chapter provides an introduction to the three main categories of approaches: Section 4.1 discusses data-level techniques, Section 4.2 cost-sensitive learning framework, and Section 4.3 algorithm-level modifications. Finally, Section 4.4 considers the distinctive challenges of data imbalance in natural language processing and approaches designed for imbalanced text classification.

4.1 Sampling Techniques

In practice, sampling methods constitute arguably the most popular category of approaches to imbalanced classification [77, 85, 89]. The two main reasons for their popularity are that, first, as they operate on the data-level, they can be accompanied with any learning algorithm, and second, they do not require assigning misclassification costs explicitly [84, 89].

Sampling can be divided into three main approaches: (i) undersampling, where some of the majority class observations are removed from the training set, (ii) oversampling, where the size of the minority class is increased in the training set, and (iii) different adaptations and combinations of (i) and (ii) (for an overview, see, e.g., [37]). The most straightforward examples of undersampling and oversampling are random undersampling (RUS) and random oversampling (ROS), where either a predetermined number of majority instances are randomly deleted, or the minority class is appended by duplicating a given number of randomly selected existing

minority observations, respectively.

The idea of sampling is that by duplicating or removing observations in the training set it effectively induces an uneven cost distribution [89]. However, there are certain issues with the naive sampling approaches. If the imbalance ratio is high, RUS has to remove a large share of the majority observations, and as this is done randomly, there is a high risk of losing important information [4, 54, 57, 87, 89]. On the other hand, with ROS, in case of severe imbalance, many of the minority observations get duplicated multiple times which makes the learning algorithm prone to overfit the training data [4, 57, 87, 89]. Due to these reasons, majority of sampling techniques applied in practice represent either informed undersampling or synthetic oversampling.

The idea of informed undersampling is to apply a set of rules and heuristics for selecting which majority observations to remove instead of doing it randomly (see, e.g., [54]). Undersampling can also be based on ensemble learning, i.e., training multiple classifiers (that vote for the final classification) on multiple different undersampled training sets in order to mitigate the risk of losing information while keeping the benefits of undersampling (see, e.g., [54, 77]). In synthetic oversampling, the minority class is not oversampled with existing observations but rather with synthetically generated instances in order to reduce the risk of overfitting. The idea of synthetic oversampling was first introduced in 2002 in a groundbreaking sampling approach, Synthetic Minority Over-sampling Technique (SMOTE) [13]. To this day, SMOTE is still arguably one of the most applied oversampling methods.

The synthetic observation generation process of SMOTE is based on considering one minority observation at a time and then selecting one or more of its k nearest (minority) neighbors and creating synthetic observations as random convex combinations with the selected neighbors [13]. That is, existing observations do not get duplicated with SMOTE, and, consequently, the risk of overfitting is reduced compared to ROS [57]. However, researchers have also pointed out certain limitations related to SMOTE that may hinder the generalization of minority class. In SMOTE, synthetic observations are generated uniformly across the minority sample and are also always bounded by the convex hull of the minority class. In addition, SMOTE may in some cases be sensitive to the hyperparameter k and prone to generate noisy synthetic observations. [6, 24, 37].

Adaptive Synthetic Sampling (ADASYN) [36] is a popular modification of SMOTE aiming to produce better generalization of the minority class. The number of synthetic observations that ADASYN generates in the neighborhood of each minority observation depends on the proportion of how many of its all k nearest neighbors belong to the majority class. That is, ADASYN focuses sampling onto areas in the data space where there are a larger share of majority instances in order to emphasize these border

areas of the minority sample which are generally more difficult to learn in classification.

Like SMOTE and ADASYN, many popular sampling techniques are non-parametric, i.e., they do not require strong distributional assumptions. However, there exists also probabilistic sampling methods (see, e.g., [14] and [19]). On the other hand, oversampling approaches can be divided into convex and non-convex methods. For instance, SMOTE and ADASYN are convex sampling techniques as the synthetic observations are generated within the convex hull of the minority class. On the contrary, Geometric SMOTE [24] and Localized Random Affine Shadowsampling (LoRAS) [6] are examples of methods that can create non-convex synthetic observations.

There may also be certain limitations and issues related to approaching imbalanced classification with sampling. Even though there is no need to assign misclassification costs explicitly in sampling, the oversampling (or undersampling) ratio must still be selected, which implicitly poses some imbalanced cost distribution that, in turn, may not be traceable [89]. In addition, in some cases, sampling methods may be prone to cause overfitting [87].

4.2 Cost-Sensitive Learning

If all the costs $C(i, j, x)$ and posterior probabilities $P(j|x)$ of observation x belonging to class j are known, the theoretical Bayes-optimal decision rule is to assign an observation x to the class that minimizes the conditional expected cost, i.e., Bayes risk [26, 90]. That is, observation x is assigned label l such that

$$l = \arg\min_i \sum_j P(j|x)C(i, j, x).$$

It is also possible to, on top of the known classes, include an "uncertain" class to the decision making process. That is, if the given observation is too challenging to label for the classifier (i.e., the classification cannot be made with a predetermined certainty), it can be passed on to the attention of an expert with a fixed cost. [80]

Technically, cost-sensitive learning and classification can be performed with any posterior probability estimation approach, but here we focus on applying machine learning algorithms for making cost-sensitive decisions. In practice, the challenge of cost-sensitive learning with machine learning is that, although many standard machine learning classification algorithms produce some score describing the certainty of each classification, typically, these scores cannot be interpreted as accurate probability estimates [23, 25, 90, 91]. For instance, SVM gives a certainty score based on the distance between the given observation and the class-separating margin. Even

though the scores could be normalized into the range of $[0, 1]$, these scores typically are not accurate probability estimates, as the distances from the margin are not proportional to the conditional probabilities. [91]

In a decision tree, a certainty score of a leaf can be computed as $\frac{k}{n}$, where k is the number of correctly classified instances and n is the total number of instances in the leaf. However, these certainty scores can rarely be used as accurate probability estimates, as the learning algorithms typically aim to make the leaves as homogeneous as possible and as, especially with imbalanced data, some leaves may contain only a small number of observations. [90] On the other hand, naive Bayes classifier technically calculates the scores as posterior probability estimates, but, as the feature independence assumption of naive Bayes very rarely actually holds in reality, these estimates are typically too extreme [23, 90].

In general, standard classification algorithms are poor at estimating posterior probabilities—as it is not what they have been designed to do—but are typically accurate at ranking observations based on the conditional probabilities [25, 84, 90]. Thus, researchers in the field of cost-sensitive learning have focused on how to produce calibrated probability estimates based on the scores (and rankings) that the classifiers output, for instance, with decision trees and naive Bayes classifier [90], with naive Bayes and SVM [91], and with SVM [25].

The decision tree probability estimates can be improved, for example, with smoothing. This can be done by including pseudo-observations in both (or all) classes in the given leaf when computing the estimate. However, standard Laplace smoothing adjusts all estimates towards $\frac{1}{N}$ (where N is the number of classes), whereas, from Bayesian perspective, the conditional probability estimates should be shrunken towards the marginal probabilities. The estimates can be smoothed towards the base rates, e.g., by m-estimation, which replaces the probability estimate $\frac{k}{n}$ by $\frac{k+b\times m}{n+m}$, where b is the base rate of the class and m is a hyperparameter controlling how strongly the estimates are smoothed. In order to reduce the variance of the estimates, instead of the leaves, the estimates can be computed by using parent (or grandparent) nodes that include a higher than some predetermined number of observations. [90]

Other estimation approaches include, for instance, bootstrap aggregation (bagging) and binning. The idea of bagging is to train an ensemble of classifiers on bootstrap samples and to aggregate the classifications by voting in order to improve the classification stability. Bagging can also be used for estimating the probabilities $P(j|x)$ by taking the share of classifiers that vote class j for observation x (see, e.g., [22]). Yet, studies have argued that bagging is not suited for probability estimation and, in general, does not produce unbiased estimators, as the voting scores in bagging inherently measure the stability of the base learner for a given observation and not the conditional probability of a certain class [59, 90].

The idea of binning is to rely on the accuracy of classifiers in ranking the instances. The range of certainty scores (and respective training observations) are divided into bins. The probability estimate of a bin for class j is computed as the number of class j training observations in the bin divided by the total number of observations in the bin. That is, binning gives a discrete step-wise function of probability estimates, where the number of different estimates is limited by the number of bins. However, the number of bins should be small enough to reduce the variance of the binned estimates. [25, 90, 91]

In addition to probability estimation, classification costs are rarely explicitly known and have to be estimated as well [70, 90]. Usually the only available approach is to use the training data for predicting future costs, e.g., by (multiple) linear regression. There are, however, certain issues related to predicting costs based on training data. First, explicit costs are not always known even in the training set. Second, even if the costs are known, they usually depend not only on the observation itself but also on its realized class—or it can be that the costs are only available for observations in a certain class. [90]

This phenomenon is known as the sample selection bias. For example, in a data set which includes information of people who were solicited to donate money for a charity, the costs (or, in this particular case, benefits) are only known for the positive class, i.e., people who did donate. Yet, imputing zero costs for negative observations in estimation is clearly wrong, as it would be similar to assigning a zero donation probability for every observation in the negative class. [90]

The problem with sample selection bias is that, when the costs are observation-dependent, there are two underlying (correlated) probabilistic processes: one determining the class of the observation and another determining the cost(s) related to the given observation. Whether the cost(s) get observed, however, depends on the realized class of the observation. Another example of sample selection bias would be a data set of people who had loaned money from a bank; if a person successfully repaid their loan they have a zero cost—though, had they defaulted, the cost obviously would not have been zero. For approaches addressing the sample selection bias in cost estimation see, for example, [39] and [90].

4.3 Algorithmic Approaches

Certain learning algorithms feature a possibility to incorporate asymmetric penalties for the set of classes in the decision rule optimization or to bias the classification threshold towards the minority class by some other means. For instance, with naive Bayes classifier, this can be done by altering the class prior distribution in favor of the minority class(es). SVM

is another classifier that has been extensively considered in the context of classification with imbalanced class and cost distributions (see, e.g., [1, 45, 83], and the references therein).

Any approach that introduces new hyperparameters or utilizes the opportunity to alter existing hyperparameters to bias the decision rule in favor of the minority class can be considered to fall under the category of algorithmic approaches. However, the majority of learning algorithms do not approach classification as a probabilistic optimization problem based on explicit costs and posterior probability estimates. Therefore, a classifier achieved by an algorithmic approach is not necessarily guaranteed to agree with the theoretical Bayes-optimal decision rule. [45]

In many classification algorithms, such as SVM, it is possible to manually shift the fitted decision boundary. Classifiers often output a certainty score which is only subsequently mapped into a binary label based on a classification threshold. Typically, this threshold is by default set to "the middle", but it may also be altered to better acknowledge the imbalanced classes and misclassification costs, i.e., such that a smaller certainty score is required for a positive classification than for a negative classification. With SVM, this method is referred to as boundary movement. The decision threshold can be optimized, for example, with cross-validation. [45]

Biased penalties SVM (BP-SVM) [83] is one of the extensions to SVM that alter the hyperparameters to put more emphasis on the minority class. In BP-SVM, the regularization hyperparameter C is split into two penalty terms, C_+ for the positive class ($y = 1$) and C_- for the negative class ($y = -1$). That is, the sums of hinge losses of positive and negative observations are weighted by C_+ and C_-, respectively, transforming the optimization problem into

$$\begin{aligned} \arg\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C_+ \sum_{j|y_j=1} \xi_j + C_- \sum_{j|y_j=-1} \xi_j \\ \text{s.t.} \quad & y_j(w^\top x_j + b) \geq 1 - \xi_j, \ \forall j, \\ & \xi_j \geq 0, \ \forall j. \end{aligned}$$

4.4 Data-Specific Challenges: Imbalanced Text Data

4.4.1 Modeling Natural Language

Significance of machine learning based natural language processing in data mining and analytics has increased massively in the recent decades. The applications of machine learning and statistical text mining and analysis include, e.g., information retrieval, machine translation, generative

language models, and text classification. Natural language differs notably from many other data domains, and thus modeling natural language is a critical—and not trivial—first step when working with text data. [51, 53]

The approaches for transforming text data to numerical form can be broadly divided into document level and word (or n-gram or character n-gram) embedded approaches. In document level approaches, an observation (i.e., a text document) is represented with a (fixed) length vector, whereas a word embedding model maps words to (fixed) length vectors and a document is represented either as a matrix or as a sequence of vectors. Some approaches do not consider words as the base units but rather word or character n-grams. A word n-gram model splits a text to units of n consecutive words. For example, a 2-gram (bigram) embedding of the sentence "this is an example" would be ("this is", "is an", "an example"). On the other hand, a character n-gram model splits a text to units of n consecutive characters. [51]

The process of extracting the base units from a text is called tokenization—and correspondingly, the units of text are commonly referred to as tokens. Tokenization can, for instance, be based on splitting a text using white spaces and removing the punctuation marks. If the tokens are words, some or all of their conjugations may also be removed. This is referred to as word normalization, and it can be done, e.g., by stemming or lemmatization. Both of these approaches aim at mapping words to simpler form such that the feature space is reduced and words with similar meanings actually look exactly the same. [51]

Word normalization is helpful in modeling approaches where the tokens are treated as features; as there is not a measure of similarity between a pair of tokens, they either are the same feature or not. Lemmatization finds the morphological root of a word, whereas stemming finds a "body" or a "stem" of a word. [51] The stem itself does not necessarily have to be a word, and there exist multiple approaches for stemming—probably the most used of them being the Porter stemmer [68].

On top of tokenization and word form normalization, *uninformative* tokens are often removed—this process is commonly called stopword removal. Stopwords are typically considered to consist of highly frequent tokens (such as articles and prepositions) that possess only little semantic information and can in some cases be treated as noise in the data (see, e.g., [73]). Though, it is important to note that stopwords depend on the language and also commonly on the context [32].

In general, word frequencies and their ranking based on the frequencies exhibit a power law type relationship (commonly known as Zipf's law [93]). That is, a majority of a text sample consists of the most frequent tokens, and thus, removing the most common tokens can help the classifier focus on the more informative tokens. In addition, tokens with very low frequencies in the sample are sometimes also considered to be "stopwords"

that are also removed. The process of pruning less informative words from the vocabulary may also be seen as a form of feature selection [32].

An example of Zipf's law is presented in Figure 4.1, where the frequencies of distinct words appearing in the first chapter of this thesis have been plotted against their respective ranking. On the left hand side, the figure presents the distribution of the total vocabulary (after preprocessing) consisting of 428 different words, and on the right hand side, it presents only the top 20 most frequent words. The most frequent word in the chapter is *the* (as it typically is in any English language text) which appears in total 55 times. A bit over half of the text in Chapter 1 consists of the 48 most frequent words, which accounts for only about 11% of the total vocabulary.

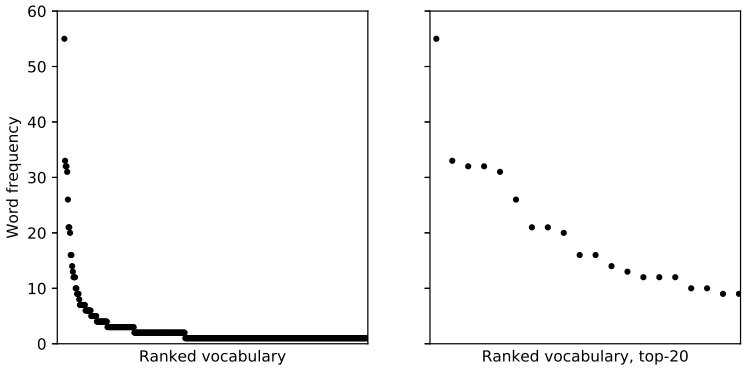

Figure 4.1. Word frequency distribution of the first chapter.

After preprocessing, text documents need to be embedded into numerical form in order to be digestible for statistical and machine learning algorithms. Though, it should be noted that classification can also be done without machine learning algorithms, for instance, by labeling documents based on a sentiment dictionary (see, e.g., [10]). Text can be transformed into numerical form, for example, by embedding tokens into vectors with, e.g., one-hot encoding or semantic word vector models, such as word2vec [61, 62] or fastText [9, 50] algorithms. The semantic word vector models are trained such that the embeddings they output capture some information about what the words mean, such that words with similar meanings have similar embeddings.

Besides word embeddings, another approach is to transform text documents straight into vectors. Arguably the most popular approach for embedding documents is a bag-of-words model, where each token in the training set is assigned an index and a document is represented as a vector with a length equal to the size of the vocabulary. Each index in a bag-of-words vector represents the frequency of that token in the given document. The benefit of semantic word vectors is that they offer a similarity measure

between words, whereas the benefit of the bag-of-words model is that it is easy to implement for any data set and for any language without the need for a pre-trained embedding model. [51]

On the other hand, the bag-of-words model loses information about word order as it treats a document just as a set of tokens. To combat the drawbacks of a bag-of-words model, the bag-of-words vectors are often transformed before fitting a classifier, for instance, with term frequency - inverse document frequency (TF-IDF) transformation (see, [74]). Moreover, in some cases, augmenting TF-IDF with a transformation based on document length and token weight normalization may improve classification (see, [71]).

Modeling approaches that transform text documents into fixed length vectors enable the use of any general-purpose classification algorithm—naive Bayes and SVM, for instance, being popular choices (see, e.g., [23, 71]). On the other hand, deep neural networks that process text as a sequence of embedded word vectors have become increasingly popular in natural language processing in general and also in text classification. These approaches include, for example, recurrent neural networks (with long short-term memory (LSTM) [43] and gated recurrent unit (GRU) [15] extensions) and, the current state-of-the-art method in large language models, the transformer architecture [82]. On top of machine learning algorithms, probabilistic models, such as latent Dirichlet allocation [8] and hidden Markov model, may also be used in text classification. For more on approaches for text preprocessing and classification, see, for example, [51, 53], and the references therein.

4.4.2 Imbalanced Text Classification

Just as in any other domain, data in text classification tasks may be imbalanced as well. In fact, classification of natural language is almost always imbalanced in practice, as the aim is typically to identify text documents relating to one specific topic out of all possible topics [32, 40]. The general challenges of data imbalance (discussed also in this thesis) are relevant to imbalanced text classification as well. However, the distinctive nature of text data may pose difficulties that would not be encountered in other imbalanced classification tasks apart from natural language. [14, 64]

The general approaches to imbalanced classification are usually viable also for imbalanced text classification. Random oversampling and undersampling can be used with any modeling approach as they simply duplicate or remove existing observations—though, the risk of overfitting with random oversampling and information loss with random undersampling are present also in text classification [40]. On the other hand, algorithmic approaches focus on modifying the classification algorithm or the loss function and are thus applicable as such to imbalanced text data as well.

Depending on the modeling approach, general-purpose synthetic resampling can be applied to text data as well. That is, in document-embedded models, where the observations are represented with fixed-length vectors, there is no technical reason why, say, oversampling with SMOTE could not be used (see, e.g., [14, 64]). However, in these cases, the observations are typically very sparse, which can cause issues when applying general convex oversampling (or undersampling) techniques. On the other hand, when modeling documents as sequences of word embeddings, general-purpose sampling methods may not be an option, and instead, the minority class should be oversampled by some other means, for instance, with data augmentation techniques [40].

In document-embedded approaches, the feature space consists of the tokens that appear in the training documents. Consequently, the feature space of the minority class is a subset of the total vocabulary. When using a convex oversampling method, the generated synthetic observations are guaranteed to lie within the convex hull of the minority class. That is, the set of tokens which appear in the minority sample is preserved. However, this does not reflect the property of natural language that the number of distinct words in a text sample tend to grow as a function of the sample size—a phenomenon described by Heaps' law [38] in natural language processing, and, with a slightly different formulation, by Herdan's law in linguistics [41] (see also [27]). These issues related to applying general-purpose sampling approaches to text data are discussed in detail in Publication II, where we also introduce a novel text oversampling method which takes into account the special property of text data described by Heaps' (and Herdan's) law.

Researchers have also designed oversampling approaches particularly tailored for text data. For instance, [14] propose a text oversampling method based on a Latent Dirichlet Allocation (LDA) [8] model, and [64] propose a framework for text oversampling based on a latent semantic distributional model. Text augmentation can also be used as a basis for text oversampling, with approaches encompassing a wide range of methods from simple synonym replacement procedures to text generation with large language models [40]. For example, [86] present a straightforward augmentation technique consisting of synonym replacements and word insertions, swaps, and deletions, and [56] propose a text oversampling method using sequence generative adversarial networks.

5. Summaries of the Articles

5.1 Negative Economic Sentiment Index Based on Finnish News Titles

The article presents a novel economic sentiment index (i.e., soft indicator) for the Finnish economy based on machine learning sentiment classification of Finnish news titles. Recently, soft indices have become increasingly important in economic forecasting and nowcasting, mostly due to the fact that they become available earlier than hard macroeconomic data and thus can help assessing the current and near-future state of the economy. The most used economic sentiment indices are typically based on surveying certain groups of people (e.g., consumers or purchase managers).

In this article, we apply supervised machine learning text classification for identifying news titles with negative economic sentiment. Our negative economic sentiment index is defined based on the monthly-aggregated proportional frequencies of negative news in the coverage. We show that there is a negative correlation between our index and the consumer confidence index by Statistics Finland, and interestingly, our index seems to lead the consumer confidence index by one month. Moreover, our index has a positive correlation with Finnish stock market volatility and a negative correlation with the rate of change of GDP. Finally, by using a simple vector autoregression (VAR) model, we evaluate how certain other macro variables respond to changes in the level of negative economic sentiment.

5.2 Extrapolated Markov Chain Oversampling Method for Imbalanced Text Classification

The article introduces a novel text oversampling technique for considering certain distinctive properties of text data that general oversampling methods are unable to address. In particular, many general-purpose over-

sampling methods generate synthetic observations as (random) convex combinations of subsets of existing minority class training examples. The issue with applying this general approach to natural language data is that, when the sample size of text increases, the feature space (i.e., the number of distinct words) typically also grows as described by Heaps' law. However, generating synthetic observations as convex combinations of minority class observations clearly preserves the feature space as immutable.

Our approach in this article is based on an assumption that the structure of text can be modeled by dividing it into two properties: topic and sequence. Furthermore, we assume that the inherent sequential structure of text is, to some extent, independent of the topic of text. Our method utilizes Markov chains and allows the feature space (i.e., vocabulary) of the minority class to expand in oversampling by learning the sequential structure partly from both classes, but by emphasizing the minority class in oversampling. We test our approach against prominent (both general-purpose and text) oversampling methods on multiple well-known multiclass text data sets and show that our approach outperforms the other methods in certain evaluation statistics, especially when the imbalance is severe.

5.3 On F_β-score and Cost-Consistency in Evaluation of Imbalanced Classification

Explicit misclassification costs are rarely available in practice and are often difficult to estimate. Cost estimation can also be complicated by the fact that different costs may have different units, i.e., they can be money, time, or, for example, quality of life. Thus, conventional classification evaluation measures are typically defined independent of any costs. Yet, even if the actual costs are not explicitly known, they can still exist, making evaluation of classifiers on imbalanced data highly challenging. F_β-score is often recommended as one of the go-to evaluation measures in imbalanced classification. However, some researchers have questioned whether it actually is an appropriate measure for imbalanced classification.

This article argues that, since real-life decision-making problems always have some classification costs (whether or not they are known), it would be important to understand how different evaluation measures behave in relation to total classification cost. We introduce a framework of evaluation measure cost-consistency such that an evaluation measure is called cost-consistent if decreasing total classification cost also increases the value of the considered evaluation measure. In this article, F_β-score is compared to two other cost-independent evaluation measures, informedness and balanced accuracy, under a simple cost structure. It is shown that F_β-score is not cost-consistent, whereas, under certain conditions, informedness and balanced accuracy are cost-consistent.

6. Discussion and Future Prospects

In Publication I, we introduce an economic sentiment index for the Finnish economy using machine learning classification of news headlines. The results are significant, as, not only is there a negative correlation between our index and the consumer confidence index by Statistics Finland, our index seems to lead the consumer confidence index by somewhat a month. Our index seems to also correlate with other widely used indicators measuring the current state of the Finnish economy. The strength of our approach lies in the fact that so-called soft indicators have become increasingly important in economic forecasting as they are available earlier than hard macroeconomic data, and our index seems to be provide useful information even earlier than the currently used soft indicators.

Publication II presents an oversampling method for imbalanced text classification, which takes into account one of the distinctive features of natural language data; that is, as described by Heaps' law, the vocabulary should grow as the sample size of text is increased. Our work is significant, since, as far as we know, our approach is the first to explicitly acknowledge Heaps' law in text oversampling. Moreover, the extensive empirical experiments we conduct show that our method is able to outperform other prominent sampling approaches in multiple occasions. The strength of our approach is that, as it is based on a Markov chain model and on utilizing information in the whole training data rather than just in the minority sample, it is an effective method also for small and severely imbalanced text data sets.

In Publication III, we introduce a concept of cost-consistency for assessing different (binary) evaluation measures. The main argument of the work is that, while true misclassification costs are often unknown and difficult to estimate in real life, they still do exist, and thus, it is important to understand how the applied classification evaluation measures behave in relation to different cost structures. The results are significant, as we show that one of the most used evaluation measures in (imbalanced) classification, F_β-score, is not cost-consistent for any given cost matrix in an observation-independent and linear cost system. The strength of the work

is that the cost inconsistency of F_β-score is both derived mathematically as well as illustrated empirically. In addition, we address certain alternative measures and explain the conditions for their cost-consistency.

Future prospects for Publication I include involving more news sources for the index and experimenting with more advanced text classification algorithms. The oversampling method introduced in Publication II is based on a unigram model, but it could be extended to consider, e.g., bigrams or trigrams as well. Finally, future prospects for Publication III involve including other conventional evaluation measures in to consideration as well and assessing cost-consistency under some more complex cost systems.

7. Key Terms of the Doctoral Thesis

This chapter provides concise definitions and explanations of three selected key terms of the thesis in English and Finnish. The terms are listed in the dictionary by Suomen Tilastoseura [2].

In English

1. **classification**
   Classification refers both to the process of *learning* or *fitting* a classification rule as well as to the task of using a classification rule for assigning a label from a set of classes for an observation as accurately as possible. Classification is done based on measured *features* or *variables*. The ability to use a classifier for accurate predictions is based on an assumption that, considering the measurements, observations in one class share some characteristics while observations in different classes have distinct characteristics.

2. **confusion matrix**
   A confusion matrix is a k-by-k table that summarizes the performance of a classifier on a given data set, where k is the number of classes. The labels predicted by the classifier are conventionally listed as the rows and the true labels as the columns of the table. Each entry in the table represents the number of observations that belong to a given class and are assigned a given label by the classifier. In a binary case, these values are the number of true positives, false positives, false negatives, and true negatives. Sometimes, instead of presenting the absolute counts, the entries in a confusion matrix are normalized by the row sums, by the column sums, or by the total number of observations.

3. **text mining**
   Text mining is the process of developing and using algorithms for extracting useful and structured information from natural language. Unlike

data in many other domains, text data is inherently unstructured, and thus analyzing text data requires methods specifically designed for the task. Applications of text mining and analysis include, for instance, lemmatization, part-of-speech tagging, text classification, information retrieval, sentiment analysis, and text summarization.

Suomeksi

1. **luokittelu**
   Luokittelulla tarkoitetaan sekä luokittelusäännön sovittamista dataan että sovitetun luokittelusäännön käyttämistä havaintojen luokitteluun. Luokittelu tapahtuu havainnoista mitattujen muuttujien arvojen perusteella ja perustuu oletukseen, että samaan luokkaan kuuluvilla havainnoilla on joitain yhteisiä ominaisuuksia, kun taas eri luokkien välillä havainnot ovat myös joillakin tavoin erilaisia.

2. **sekaannusmatriisi**
   Sekaannusmatriisi on k-kertaa-k taulukko, joka on yhteenveto luokittelusäännön soveltamisesta tiettyyn aineistoon, jossa on k luokkaa. Sekaannusmatriisin rivit vastaavat tyypillisesti luokittelijan ennustamia luokkia ja sarakkeet havaintojen todellisia luokkia. Taulukon arvot puolestaan kertovat kuinka moni kunkin sarakkeen luokkaan kuuluva havainto on luokiteltu kunkin rivin luokkaan. Havaintojen määrien sijaan taulukon arvot toisinaan normalisoidaan rivi- tai sarakesummilla tai havaintojen kokonaismäärällä.

3. **tekstinlouhinta**
   Tekstinlouhinnalla tarkoitetaan algoritmien kehittämistä ja hyödyntämistä jäsennellyn ja hyödyllisen tiedon poimimiseen luonnollisesta kielestä. Toisin kuin useat muut aineistot, teksti ei ole lähtökohtaisesti muodossa, jota voidaan käsitellä tilastollisilla menetelmillä. Tekstinlouhinnan ja luonnollisen kielen analyysin sovelluskohteisiin lukeutuvat mm. sanojen perusmuotoistaminen, sanaluokkien tunnistaminen, tekstin luokittelu, tiedonhaku, sentimenttianalyysi ja dokumenttien tiivistäminen.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning*, ECML 2004, pages 39–50, 2004.

[2] J. Alho, E. Arjas, E. Läärä, and P. Pere. *Tilastotieteen sanasto. Suomen Tilastoseuran julkaisuja no. 8. 2. laitos*. Suomen Tilastoseura, 2023.

[3] G. E. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5—6):519—-533, 2003.

[4] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[5] G. E. Batista, R. C. Prati, and M. C. Monard. Balancing strategies and class overlapping. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, IDA 2005, pages 24–35, 2005.

[6] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer. LoRAS: an oversampling approach for imbalanced datasets. *Machine Learning*, 110:279—-301, 2021.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, NY, 2006.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[10] C. Bortoli, S. Combes, and T. Renault. Nowcasting GDP growth by reading newspapers. *Economie et Statistique*, 505–506:17–33, 2018.

[11] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144—-152. Association for Computing Machinery, 1992.

[12] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20*, 2007.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[14] E. Chen, Y. Lin, H. Xiong, Q. Luo, and H. Ma. Exploiting probabilistic topic models to improve text categorization under class imbalance. *Information Processing & Management*, 47(2):202–214, 2011.

[15] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[16] P. Christen, D. Hand, and N. Kirielle. A review of the F-measure: Its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 56(3), 2023.

[17] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20:273—-297, 1995.

[18] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[19] B. Das, N. C. Krishnan, and D. J. Cook. RACOG and wRACOG: Two probabilistic oversampling techniques. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):222–234, 2015.

[20] J. Davis and M. Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML'06, pages 233–240, 2006.

[21] M. Denil and T. Trappenberg. Overlap versus imbalance. In *Proceedings of the 23rd Canadian Conference on Artificial Intelligence*, Canadian AI 2010, pages 220–231, 2010.

[22] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[23] P. Domingos and M. Pazzani. Beyond independence: Conditions for optimality of the simple Bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, ICML'96, pages 105–112, 1996.

[24] G. Douzas and F. Bacao. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences*, 501:118–135, 2019.

[25] J. Drish. Obtaining calibrated probability estimates from support vector machines. Technical report, Department of Computer Science and Engineering, University of California, San Diego, CA, 2001.

[26] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York, NY: Wiley, 1973.

[27] L. Egghe. Untangling Herdan's law and Heaps' law: Mathematical and informetric arguments. *Journal of the American Society for Information Science and Technology*, 58(5):702—-709, 2007.

[28] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, IJCAI'01, pages 973–978, 2001.

[29] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona. A survey on missing data in machine learning. *Journal of Big data*, 8, 140, 2021.

[30] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1:291–316, 1997.

[31] A. Fernández, V. López, M. Galar, M. J. del Jesus, and F. Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97–110, 2013.

[32] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[33] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference, Computational Learning Theory*, EuroCOLT'95, pages 23–37, 1995.

[34] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[35] D. Hand and P. Christen. A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28:539–547, 2018.

[36] H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[37] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[38] H. S. Heaps. *Information Retrieval: Computational and Theoretical Aspects*. Academic Press, Inc., 1978.

[39] J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–161, 1979.

[40] S. Henning, W. Beluch, A. Fraser, and A. Friedrich. A survey of methods for addressing class imbalance in deep-learning based natural language processing. *arXiv preprint arXiv:2210.04675*, 2023.

[41] G. Herdan. *Quantitative Linguistics*. London: Butterworths, 1964.

[42] T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282, 1995.

[43] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735—-1780, 1997.

[44] R. C. Holte, L. E. Acker, and Porter B. W. Concept learning and the problem of small disjuncts. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, IJCAI'89, 1989.

[45] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos. Cost-sensitive support vector machines. *Neurocomputing*, 343:50–64, 2019.

[46] K. Jackowski, B. Krawczyk, and M. Woźniak. Cost-sensitive splitting and selection method for medical decision support system. In *Proceedings of the 13th International Conference, Intelligent Data Engineering and Automated Learning*, IDEAL 2012, pages 850–857, 2012.

[47] N. Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence*, ICAI, pages 111–117, 2000.

[48] N. Japkowicz. Concept-learning in the presence of between-class and within-class imbalances. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Canadian AI 2001, pages 67—77, 2001.

[49] T. Jo and N. Japkowicz. Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6(1):40–49, 2004.

[50] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[51] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Online manuscript released January 12, 2025. Available at `web.stanford.edu/~jurafsky/slp3/`, 3rd edition, 2025.

[52] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.

[53] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. Text classification algorithms: A survey. *Information*, 10(4), 150, 2019.

[54] X. Y. Liu, J. Wu, and Z. H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

[55] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019.

[56] Y. Luo, H. Feng, X. Weng, K. Huang, and H. Zheng. A novel oversampling method based on SeqGAN for imbalanced text classification. In *Proceedings of 2019 IEEE International Conference on Big Data*, pages 2891–2894, 2019.

[57] V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

[58] V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Systems with Applications*, 39(7):6585–6608, 2012.

[59] D. Margineantu. On class probability estimates and cost-sensitive evaluation of classifiers. In *Workshop Notes, Workshop on Cost-Sensitive Learning, International Conference on Machine Learning*, 2000.

[60] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115—-133, 1943.

[61] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[62] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[63] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning (Second Edition)*. The MIT Press, 2018.

[64] A. Moreo, A. Esuli, and F. Sebastiani. Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'16, pages 805–808, 2016.

[65] B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67—-97, 1989.

[66] M. Núñez. The use of background knowledge in decision tree induction. *Machine Learning*, 6:231—-250, 1991.

[67] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[68] M. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[69] D. M. W. Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[70] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203—-231, 2001.

[71] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive Bayes text classifiers. In *Proceedings of the 20th International Conference on Machine Learning*, ICML'03, pages 616–623, 2003.

[72] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386—-408, 1958.

[73] H. Saif, M. Fernández, Y. He, and H. Alani. On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, LREC 2014, pages 810—-817, 2014.

[74] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[75] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571–595, 2014.

[76] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[77] M. A. Tahir, J. Kittler, K. Mikolajczyk, and F. Yan. A multiple expert approach to the class imbalance problem using inverse random under sampling. In *Proceedings of the 8th International Workshop, Multiple Classifier Systems*, MCS 2009, pages 82–91, 2009.

[78] A. M. Turing. Computing machinery and intelligence. *Mind*, 59 (236):433—-460, 1950.

[79] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, pages 369–409, 1994.

[80] P. D. Turney. Types of cost in inductive concept learning. *arXiv preprint arXiv:cs/0212034*, 2002.

[81] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2), 1971.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[83] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60, 1999.

[84] S. Viaene and G. Dedene. Cost-sensitive learning and decision making revisited. *European Journal of Operational Research*, 166(1):212–220, 2005.

[85] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski. On the class overlap problem in imbalanced data classification. *Knowledge-Based Systems*, 212, 106631, 2021.

[86] J. Wei and K. Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389. Association for Computational Linguistics, 2019.

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

[88] G. M. Weiss. The impact of small disjuncts on classifier learning. In *Data Mining: Special Issue in Annals of Information Systems, vol 8*, pages 193–226. Springer, Boston, MA, 2010.

[89] G. M. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proceedings of 2007 International Conference on Data Mining*, DMIN'07, 2007.

[90] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'01, pages 204–213, 2001.

[91] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'02, pages 694–699, 2002.

[92] S. Zhang, Z. Qin, C. X. Ling, and S. Sheng. "Missing is useful": Missing values in cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1689–1693, 2005.

[93] G. K. Zipf. *Human behavior and the principle of least effort*. Addison-Wesley Press, Cambridge, MA, 1949.

Business, Economy

Art, Design, Architecture

Science, Technology

Crossover

Doctoral Theses

Aalto DT 242/2025

ISBN 978-952-64-2863-5
ISBN 978-952-64-2862-8 (pdf)

Aalto University
School of Science
Department of Mathematics
and Systems Analysis
aalto.fi