
HUMAN-AWARE APPLICATION OF DATA SCIENCE TECHNIQUES

Bernat Coma-Puig

Supervisor: Josep Carmona



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Barcelona, January 2022

Acknowledgment

First of all, I would like to give special thanks to my supervisor Josep Carmona. I will always be grateful to him for his guidance and accompaniment during my PhD. I am aware that I have been fortunate to have him as a supervisor.

I would also like to express my gratitude to Ricard Gavaldà, who shared his expertise in the field of machine learning and data mining with me.

I also thank all the researchers with whom I have worked: Massimiliano de Leoni, Riccardo Galanti, Nicolò Navarin, Luz Muñoz, Noemí Orué, Albert Calvo and Alexandra Yamaui.

I would also like to express my gratitude to Naturgy, especially to Santiago Alcoverro, Manuel Cirugeda and Enric Argelich.

And finally, I would like to mention Ernesto Diaz-Avilés, who introduced me to the world of data research. Without that experience I would probably not have ended up doing my PhD.

En primer lloc, m'agradaria donar un agraïment especial al meu supervisor en Josep Carmona. Sempre li estaré agraït per la seva orientació i acompanyament durant el meu doctorat. Sóc conscient que he tingut la sort de tenir-lo com a supervisor.

També vull expressar el meu agraïment a en Ricard Gavaldà, que ha compartit amb mi el seu alt coneixement en l'àmbit de l'aprenentatge automàtic i la mineria de dades.

També agraeixo a tots els investigadors amb qui he treballat: en Massimiliano de Leoni, en Riccardo Galanti, en Nicolò Navarin, la Luz Muñoz, la Noemí Orué, la Marta Arias, l'Albert Calvo i l'Alexandra Yamaui.

També voldria expressar el meu agraïment a Naturgy, i especialment a en Santiago Alcoverro, en Manuel Cirugeda i l'Enric Argelich.

I finalment, m'agradaria esmentar l'Ernesto Diaz-Avilés, que em va introduir en el món de la recerca de dades. Sense aquella experiència probablement no hauria acabat fent el meu doctorat.

Abstract

In recent years there has been an increase in the use of artificial intelligence and other data-based techniques to automate decision-making in companies, and discover new knowledge in research. In many cases, all this has been performed using very complex algorithms (so-called black-box algorithms), which are capable of detecting very complex patterns, but unfortunately remain nearly uninterpretable.

Recently, many researchers and regulatory institutions have begun to raise awareness of their use. On the one hand, the subjects who depend on these decisions are increasingly questioning their use, as they may be victims of biases or erroneous predictions. On the other hand, companies and institutions that use these algorithms want to understand what their algorithm does, extract new knowledge, and prevent errors and improve their predictions in general. All this has meant that researchers have started to focus on the interpretability of their algorithms (for example, through explainable algorithms), and regulatory institutions have started to regulate the use of the data to ensure ethical aspects such as accountability or fairness.

This thesis brings together three data science projects in which black-box predictive machine learning has been implemented to make predictions. In each case, we contextualize the problem and explain the use of the explanatory algorithms to guarantee the robustness and quality of the model built.

Non-Technical Losses Detection System Our first case study explains the development of an NTL detection system for the international utility company from Spain Naturgy. Despite the good results achieved detecting NTL cases (especially in certain cases in which previous approaches had very poor accuracy), we suffered many of the problems regarding the quality of the data. These problems hindered our system from achieving consistent and robust results. Once we made clear the existence of these data problems, we shifted our effort from trying to make a more complex method (to detect more complex patterns) to achieving a more interpretable method, allowing both the scientists and the stakeholders to understand the patterns learnt (and therefore the detection of biases and undesired patterns). These data-related problems are partially explained in the literature but are not tackled in any other example of the literature.

Explainable Predictive Process Monitoring This thesis also explains our collaboration with the University of Padova to provide explainability to a KPI system currently implemented by the MyInvenio company. In this case, we show that using explanatory black-box algorithms can provide robust explanations in line with the company's analysts, with less human effort. The predictive process management and the explanatory algorithms are breaking new ground, and the resulting work is a pioneer in bringing them together in the literature of business process management.

Explainable Black-Box Algorithms in Social Science The classical dichotomy of interpretable algorithms vs black-box algorithms has not existed in Social Science literature since it is mandatory to understand the relation between variables. In this thesis, we analyze, using as a reference a collaboration between the author of the thesis and the Universitat de Barcelona, if the combination of black-box algorithms with explanatory methods can provide better results (e.g., a deeper understanding of the interaction between features, or more flexibility) in Social Science projects than the classical approach of using the interpretable Regression or Decision Tree models.

The unique characteristics of each project allow us to offer in this thesis a comprehensive analysis of the challenges and problems that exist in order to achieve a fair, transparent, unbiased and generalizable use of data in a data science project. With the feedback arising from the research carried out to provide satisfactory solutions to these three projects, we aim to:

- Understand the reasons why a prediction model can be regarded as unfair or untruthful, making the model not generalizable, and the consequences from a technical point of view in terms of low accuracy of the model, but also how this can affect us as a society.
- Determine and correct (or at least mitigate) the situations that cause the problems in terms of robustness and fairness of our data.
- Assess the difference between the interpretable algorithms and black-box algorithms. Also, evaluate how well the explanatory algorithms can explain the predictions made by the predictive algorithms.
- Highlight what the stakeholder's role in guaranteeing a robust model is and how to convert a data-driven approach to solve a predictive problem into a data-informed approach, where the data patterns and the human knowledge are combined to maximize profit.

Resum

En els darrers anys s'ha incrementat l'ús de la intel·ligència artificial i altres tècniques basades en dades per automatitzar la presa de decisions a les empreses, així com per descobrir nous coneixements en recerca. En molts casos, tot això s'ha realitzat mitjançant algorismes molt complexos (els anomenats algorismes de caixa negra), que són capaços de detectar patrons molt complexos, però malauradament segueixen sent gairebé ininterpretables.

Recentment, molts investigadors i institucions reguladores han començat a conscienciar sobre el seu ús. D'una banda, els subjectes que depenen d'aquestes decisions qüestionen cada cop més el seu ús, ja que poden ser víctimes de biaixos o prediccions errònies. D'altra banda, les empreses i institucions que utilitzen aquests algorismes volen entendre què fa el seu algorisme, extreure nous coneixements, així com prevenir errors i millorar les seves prediccions en general. Tot això ha fet que els investigadors hagin començat a centrar-se en la interpretabilitat dels seus algorismes (per exemple mitjançant algorismes d'explicabilitat), i les institucions reguladores hagin començat a regular l'ús de les dades per garantir aspectes ètics com la rendició de comptes o l'equitat.

Aquesta tesi reuneix tres projectes de ciència de dades en els quals s'ha implementat l'aprenentatge automàtic predictiu de caixa negra per fer prediccions. En cada cas, contextualitzem el problema, i expliquem l'ús dels algorismes explicatius per garantir la robustesa i la qualitat del model construït.

Sistema de detecció de pèrdues no tècniques El nostre primer cas pràctic explica el desenvolupament d'un sistema de detecció de NTL (pèrdues no tècniques) per a la companyia internacional del sector de l'energia d'Espanya Naturgy. Malgrat els bons resultats obtinguts en la detecció de casos de NTL (sobretot en alguns casos en què els enfocaments anteriors tenien una precisió molt escassa), vam patir molts dels problemes de qualitat de les dades. Aquests problemes van impedir que el nostre sistema aconseguís resultats consistents i sòlids. Un cop vam ser clarament conscients de l'existència d'aquests problemes de dades, vam canviar el nostre esforç d'intentar fer un mètode més complex (amb l'objectiu de detectar patrons més complexos) a aconseguir un mètode més interpretable, que permetés tant als científics com als *stakeholders* (els treballadors de l'empresa)

entendre els patrons apresos (i, per tant, la detecció de biaixos i patrons no desitjats). Aquests problemes relacionats amb les dades s'expliquen parcialment a la literatura, però no s'aborden en cap altre exemple de la literatura.

Monitorització de processos predictius explicables Aquesta tesi també explica la nostra col·laboració amb la Universitat de Pàdua per donar explicabilitat a un sistema KPI (Key Performance Indicator) implementat actualment per l'empresa MyInvenio. En aquest cas mostrem que l'ús d'algoritmes explicatius de caixa negra pot proporcionar explicacions sòlides en línia amb els analistes de l'empresa, amb menys esforç humà. Tant la gestió de processos predictius com els algorismes explicatius estan obrint nous camins, i el treball resultant és pioner a reunir-los en la literatura de gestió de processos empresarials.

Algorismes de caixa negra explicables en ciències socials La dicotomia clàssica d'algorismes interpretables vs. algorisme de caixa negra no ha existit a la literatura de ciències socials, ja que és obligatori entendre la relació entre variables. En aquesta tesi analitzem, utilitzant com a referència una col·laboració entre l'autor de la tesi i la Universitat de Barcelona, si la combinació d'algorismes de caixa negra amb mètodes explicatius pot donar millors resultats (p. ex., una comprensió més profunda de la interacció entre variables, o més flexibilitat) en projectes de ciències socials que l'enfocament clàssic d'utilitzar els models interpretables de regressió o arbre de decisions.

Les característiques singulars de cada projecte ens permeten oferir en aquesta tesi una anàlisi exhaustiva dels reptes i problemes que existeixen per tal d'aconseguir un ús just, transparent, imparcial i generalitzable de les dades en un projecte de ciència de dades. Amb el *feedback* derivat de la recerca realitzada per donar solucions satisfactòries a aquests tres projectes, pretenem:

- Comprendre les raons per les quals un model de predicció es pot considerar injust o fals, fent que el model no sigui generalitzable, i les conseqüències des d'un punt de vista tècnic en termes de poca precisió del model, però també com això ens pot afectar com a societat.
- Determinar i corregir (o almenys mitigar) les situacions que causen els problemes en termes de robustesa i equitat de les nostres dades.
- Avaluar la diferència entre els algorismes interpretables i els algorismes de caixa negra. També, avaluar fins a quin punt els algorismes explicatius poden explicar les prediccions fetes pels algorismes predictius.

- Destacar quin és el paper de les parts interessades (*stakeholders*) per garantir un model robust i com convertir una aproximació basada només en les dades per resoldre un problema predictiu en una aproximació que faci ús de les dades però que també es complementi amb altres coneixements, on els patrons de dades i el coneixement humà es combinen per maximitzar els beneficis.

Contents

I	Introduction	1
1	Trustworthiness in Data Science and Artificial Intelligence	3
1.1	Towards a Non-Abusive and Human-Aware Use of Data . . .	3
1.1.1	Democratisation of Data Science	3
1.1.2	Automated Data Science: Can We Trust it?	5
1.2	Application Areas and Published Work	8
1.2.1	Case of Study	8
1.2.2	Our Published Work	9
2	Preliminaries	12
2.1	Supervised Predictive Models	12
2.1.1	Data Extraction and Pre-Processing	12
2.1.2	Model Selection, Parameter Tuning and Loss Function	14
2.1.3	Deployment of the System and Post-Analysis	19
2.1.4	Challenges in Supervised Predictive Models	20
2.2	Model Transparency and Explainability	22
2.2.1	Interpretable Algorithms	23
2.2.2	Black-Box Algorithms	25
2.3	Explainable AI	28
2.3.1	XAI goals	29
2.3.2	XAI Approaches	30
2.4	Discussion	33
II	Human-Aware NTL Detection	34
3	Preliminaries in Non-Technical Losses Detection	36
3.1	Context of the Application Area	36
3.2	Related Work	37
3.3	NTL Detection: Baseline Approach	38
3.3.1	Data Processing	38
3.3.2	Creating a Classification Problem	41
3.3.3	The Process	46
3.3.4	Learning from Feedback	47

<i>Contents</i>	10
3.3.5 Algorithmic Details	48
3.3.6 Initial Results	48
3.4 Exploiting the Classification Approach	51
3.4.1 More Labeled Instances	51
3.4.2 Segmentation of the Campaigns	52
3.4.3 New Features	54
3.4.4 Algorithm and Metrics	57
3.4.5 More Results	57
3.5 Overall Analysis of the Classification Supervised Approach	58
3.6 Discussion	60
4 A Regression Approach to NTL Detection	62
4.1 System Goals and Challenges	62
4.2 The Regression Approach for NTL Detection	65
4.2.1 From Classification to Regression in NTL Detection	65
4.2.2 Experiments: Classification vs. Regression Bench-	
marking in Real Data	68
4.3 Analysing NTL Detection Beyond Benchmarking	71
4.3.1 Classification vs. Regression in Terms of Explainability	72
4.3.2 Experiments: Classification vs. Regression Explain-	
ability in Real Data	74
4.3.3 Customer Selection Through Local Explainability . .	82
4.4 Discussion	84
5 Introducing the Human Perspective	85
5.1 What is my System Learning?	85
5.2 Our Experience Using Explanatory Algorithms	87
5.2.1 The Starting Point: Statistical Analysis	87
5.2.2 Feature Importance	88
5.2.3 Local Surrogate Models	89
5.2.4 SHAP	92
5.2.5 Comparison in our NTL Detection System	95
5.3 Discussion	96
6 Human-in-the-Loop Approach to Improve NTL Detection	98
6.1 Mitigating the Existing Problems for each Model Built . . .	98
6.2 Human-in-the-Middle Approach to Implement Specific So-	
lutions for each Domain	99
6.2.1 The Proposal	99
6.2.2 The Human Analysis in the Building Process	99
6.3 A case Study with a Real Dataset	101
6.3.1 Preliminaries	101
6.3.2 Tests	103
6.4 Discussion	110

III Explainable AI in Other Fields 112

7 Explainable Predictive Process Monitoring 114

- 7.1 Business Process Management (BPM) 114
 - 7.1.1 Introduction 114
 - 7.1.2 Life Cycle 115
 - 7.1.3 Predictive Process Monitoring 117
 - 7.1.4 Predictive Process Monitoring through Machine Learning 118
- 7.2 Case Study: Explainable Predictive Process Monitoring . . 119
 - 7.2.1 Preliminaries, Context and Problem Statement . . . 119
 - 7.2.2 Case Study Analysis 120

8 Explainable Black-Box Algorithms in Social Science 126

- 8.1 Data in Social Science 126
 - 8.1.1 Data Modelling in Social Science 126
 - 8.1.2 Challenges and Possibilities of XAI in Social Science 127
- 8.2 Case Study: How NGOs Prioritize Foreign Aid Recipient Countries 128
 - 8.2.1 Preliminaries, Context and Problem Statement . . . 128
 - 8.2.2 Case Study Analysis 133

IV Conclusions and Further Research 141

9 Discussion 143

- 9.1 This Work: Contributions and Results 143
- 9.2 General Discussion 144
 - 9.2.1 The Importance of Data in Predictive Modeling . . . 144
 - 9.2.2 Involving the Stakeholder: Causality Validation and Benchmarking 145
 - 9.2.3 Interpretable vs Explained Black-Box Algorithms . . 145

Bibliography 147

Part I

Introduction

In recent years there has been an increase in the use of artificial intelligence and other data-based techniques to automate decision making in companies, as well as to discover new knowledge in research. In many cases all this has been performed using very complex algorithms (so-called black-box algorithms), which are capable of detecting very complex patterns, but unfortunately remain nearly uninterpretable.

Recently, many researchers and regulatory institutions have begun to raise awareness of their use. On the one hand, the subjects who depend on these decisions are increasingly questioning their use, as they may be victims of biases or erroneous predictions. On the other hand, companies and institutions that use these algorithms want to understand what their algorithm does, to extract new knowledge, as well as to prevent errors and improve their predictions in general. All this has meant that researchers have started to focus on the interpretability of their algorithms (for example through explicability algorithms), and regulatory institutions have started to regulate the use of the data to ensure ethical aspects such as accountability or fairness.

This thesis brings together three Data Science projects in which black-box predictive machine learning has been implemented to make predictions. In each case, we contextualize the problem, and explain the use of the explanatory algorithms to guarantee the robustness and quality of the model built.

Chapter 1

Trustworthiness in Data Science and Artificial Intelligence

1.1 Towards a Non-Abusive and Human-Aware Use of Data

1.1.1 Democratisation of Data Science

The digitisation of our society, where everything has a digital trace, has been an excellent opportunity for the companies that have started to focus on exploiting this vast amount of data for their own competitive advantage. In addition, public institutions have seen the value of exploiting the data available to provide a better service. A consequence of this is the coining of the term Data Science [134, 104], a term that encompasses a wide range of statistical analysis, data mining techniques and artificial intelligence algorithms whose focus is to process and discover new knowledge from data properly (see Figure 1.1).

Data science techniques have been prevalent during the last few years, going from being used in specific early adopters in IT companies and academia to being popular in any industry. Thus, data science is present in many fields in our daily lives, consciously or unconsciously. Below we exemplify the existence of data science solutions in three different fields:

Healthcare Using smart techniques in healthcare is not new. A clear example of this is MYCIN [124], a rule-based expert system from the 70s that could accurately diagnose and propose treatment for infections caused by bacteria such as meningitis. The system was accurate from a medical perspective, but the difficulty of integrating it in the clinical workflows (due to the state of the technology in the 70s) prevented their adoption in clinical practice. However, this system (and other rule-based chemical and medical systems, e.g. [51]) proved the possibilities of AI-based systems, and

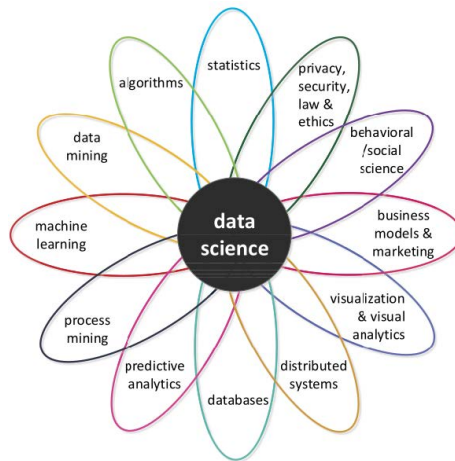


Figure 1.1: The term Data Science includes a wide range of techniques [134].

during the last 50 years there has been a great deal of work to provide AI solutions in healthcare, such that we currently possess different AI-based solutions in different fields, for instance in computer-aided interpretation of medical images [50] or heart-sound analysis [115].

That said, if we have to talk about the implementation of AI in medicine, it is necessary to talk about Watson. Watson is an IBM’s artificial intelligence clinical diagnostic decision tool that aims to support the doctor in diagnosing and treating diseases by exploiting data and knowledge from the medical literature and giving insights that would help the doctors make decisions. With this help, the doctors could be up to date on all the existing techniques and medical publications worldwide.

Beyond the detection of diseases, there exist other works that focus on prevention. For instance, data science can be used to predict populations at risk of suffering particular diseases [106], as well as to predict hospital readmission [92]. Regarding the optimization of the resources, Process Mining, the branch in Data Science that optimizes processes analyzing event logs, has successfully produced tools to improve the processes in hospitals [82].

Finally, we have recently seen with the Covid-19 virus pandemic how big data can help to provide a global vision of how the epidemic has evolved and the optimal approach to face it [9].

Finances The finance sector is one of the industries that have been using Data Science techniques to cope with different financial tasks.

In risk management institutions evaluate customers’ creditworthiness by scoring them based on their characteristics, such as their payment history or the types of credits used by the customer. The classical approach to compute this credit score is a mathematical/statistical model, but recent

approaches (e.g. [77]) advocate for the use of machine learning techniques to provide a much more complex and accurate prediction.

Financial institutions also use data to implement fraud detection algorithms that guarantee the protection of their customers against malicious credit card purchases use. As doing this analysis humanly is not feasible, we already have examples of the use of neural networks in the 1990s (e.g. [60]).

Another implementation of smart techniques is algorithmic trading, i.e., algorithms that analyze real-time financial instruments (e.g. stocks or commodity markets) to forecast market opportunities. The advantage over a human trader is both the capacity of analysing large amounts of data with no emotional factors and also the speed in making trading decisions [98].

Justice Unlike the previous examples the application of data science techniques in justice is at a very early stage.

The most groundbreaking proposal is the possibility of replacing a court judge using artificial intelligence, an idea that is not new [38]. Although it might be seen as purely science fiction, it is feasible that predictive algorithms could be helpful to complement the point of view of the judge, for instance, to analyze the possibility of re-offending and, therefore, adapt the prison sentence to this evidence-based analysis.

From a more administrative point of view, algorithms can help to reduce judges' workloads, especially in those regions where the courts are overburdened. An example of this is China's intelligent court system [144] that aims to modernize and automatize different aspects of court activities.

These examples from three different fields demonstrate the promise in terms of benefits for society with the use of data science in the big data age, either by discovering new unknown knowledge in the data, or by automating routine processes.

1.1.2 Automated Data Science: Can We Trust it?

The fact that more and more data is available has led data science to use more and more supervised algorithms from machine learning, the branch of artificial intelligence that autonomously learns mathematical models from data to automatize processes (and make predictions) with minimal human intervention. However, the strengths of using machine learning to automate processes are accompanied by several technical and ethical concerns that prevent a more extensive use of these algorithms. Here are some examples of these concerns also exemplified in Healthcare, Finances and Justice:

Healthcare The use of autonomous techniques in healthcare raises several issues in accountability, transparency, permission and privacy [40].

If a predictive cancer system (e.g., a system used to detect cancer in image analysis and suggest the optimal treatment) detects cancer and suggests a treatment for this cancer, the patient would like to understand how the algorithm made the prediction, especially if the prediction made by the system is that the disease is incurable and only palliative treatment should be provided. Similarly, the accountability of the algorithm would be complicated if the algorithm made a wrong prediction, and that might make it difficult for the patient to allow an algorithm to decide his or her treatment.

A clear example of the difficulties in exploiting large amounts of data in medicine can be seen in IBM's Watson. As we previously explained in Section 1.1.1, although it looked like a great opportunity for medicine, and IBM had partnerships with many hospitals in the USA, the results have been fair [128], and even in some cases its predictions were incorrect [114]. This is justifiable due to the difficulty of understanding ambiguities, subtleties and nuances from medical texts. These unsuccessful results (especially considering the hopes placed in the system) changed the original idea of IBM Watson replacing professional doctors to a more moderate and realistic purpose of assisting them.

Finances According to the Home Mortgage Disclosure Act data from 2017¹, 7.9% of non-Hispanic white applicants, 10.1% Asian applicants, 13.5% of Hispanics applicants and 19.3% of black applicants for a conventional mortgage were denied. Apart from the fact that there might be a human racial bias in the home mortgage approval, the automation of this process using predictive systems might replicate these biases, leading to a white applicant being more likely to being accepted than a black applicant, even when their characteristics regardless of race (e.g., income) are the same.

The problems related to the lack of interpretability are evidenced in [1], a web article from the FICO company (responsible for the FICO's credit score) that analyses the use of machine learning techniques in their credit score. FICO's results were fair: they could only increase 2% their predictive lift, while the new approach would have reduced transparency, being hard to explain to consumers and regulators their predictions. The trade-off between the improvement in metrics and the lack of interpretability did not justify using machine learning techniques.

Justice In many cases the law is interpretable, and a judge is the one who has to interpret it. That is, the judge not only determines if a crime (or, for instance, a fraud) is committed, but also the punishment, e.g., if

¹ Available at https://files.consumerfinance.gov/f/documents/bcftp_hmda_2017-mortgage-market-activity-trends_report.pdf

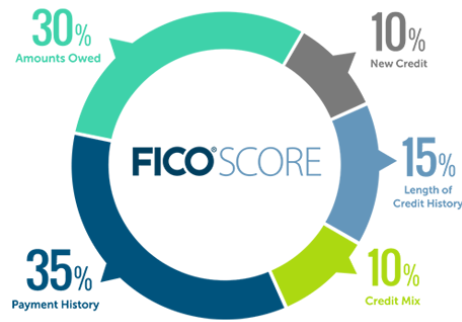


Figure 1.2: FICO is one of the most important analyst companies in terms of creditworthiness, which evaluates the customer according to the five parameters from the image. According to their webpage [1], they are aware of the bias and data-related problems that can make the prediction unfair and, therefore, they avoid the blind use of machine learning models with no interpretability and the use of conflictive feature such as customer’s race.

the accused is guilty, and if they are found guilty if they have to go to prison and the duration of the sentence.

An example of this is the GENEQUAL project² from the University of Barcelona that analyses the reasons why a judge approves a restraining order. In general, the proportion of approved restraining orders should be similar throughout Spain (with maybe slight differences between regions due to socioeconomic reasons), but that is not the case. For instance, in Martorell (a city just outside Barcelona) the probability of getting a restraining order is only 22%, while in Sant Feliu de Llobregat (another city adjoining Barcelona) the probability increases to 66%. This research analyses if the judge (or the prosecutor) can influence this result (e.g., if female judges usually accept more restraining order than male judge, or similarly if female prosecutors also increase the likelihood of approval).

This example indicates that the law (and its interpretation) is not rigid, and therefore, an autonomous system would be highly different depending on the data used: the existing prejudices, singular interpretations of the law, or political and religious opinions influence their sentences and, therefore, would also influence the autonomous system. If a predictive system is trained with harsh sentences, the resulting system could be biased to long sentences. This would be aggravated in situations where courts are influenced politically in war scenarios or political conflicts.

These examples show that the implementation of autonomous models

² SR19-0208 “GENEQUAL-The Political Economy of Gender and Inequality in the Spanish Judiciary”, from “la Caixa” Social Research Call 2019

to predict is, in certain cases, not feasible or, at least, very challenging. On the one hand, the data available to extract patterns may be biased and therefore might not correctly represent reality. Therefore, the predictive system might reproduce these biases in the prediction. On the other hand, even though the system is robust and accurate, their use would raise issues that discourage their use.

1.2 Application Areas and Published Work

1.2.1 Case of Study

This thesis brings together three Data Science projects in which the correct use of data has been a key aspect of their development.

Non-Technical Losses Detection System Our first case study (Part II) explains the development of an NTL detection system for the international utility company from Spain (Naturgy). In spite of the good results achieved detecting NTL cases (especially in certain cases in which previous approaches had very poor accuracy), we suffered many of the problems regarding the quality of the data. These problems hindered our system from achieving consistent and robust results. Once we made clear the existence of these data problems, we shifted our effort from trying to make a more complex method (with the aim of detecting more complex patterns) to achieving a more interpretable method, allowing both the scientists and the stakeholders to understand the patterns learnt (and therefore the detection of biases and undesired patterns). These data-related problems are partially explained in [61] but are not tackled in any other example of the literature.

Explainable Predictive Process Monitoring In Chapter 7 we explain our collaboration with the University of Padova to provide explainability to a KPI system currently implemented by the MyInvenio company. In this case we show that using explanatory black-box algorithms can provide robust explanations in line with the company's analysts, with less human effort. Both the predictive process management and the explanatory algorithms are breaking new ground, and the resulting work [58] is a pioneer in bringing them together in the literature of business process management.

Explainable Black-Box Algorithms in Social Science The classical dichotomy of interpretable algorithms vs black-box algorithm has not existed in Social Science literature since it is mandatory to understand the relation between variables. In Chapter 8 we analyze, using as a reference [95], if the combination of black-box algorithms with explanatory methods can provide better results (e.g., a deeper understanding of the interaction

between features, or more flexibility) in Social Science projects than the classical approach of using the interpretable Regression or Decision Tree models.

The unique characteristics of each project allow us to offer in this thesis a comprehensive analysis of the challenges and problems that exist in order to achieve a fair, transparent, unbiased and generalizable use of data in a data science project. With the feedback arising from the research carried out to provide satisfactory solutions to these three projects, we aim to:

- Understand the reasons why a prediction model can be regarded as unfair or untruthful, making the model not generalizable, and the consequences from a technical point of view in terms of low accuracy of the model, but also how this can affect us as a society.
- Determine and correct (or at least mitigate) the situations that cause the problems in terms of robustness and fairness of our data.
- Assess the difference between the interpretable algorithms and black-box algorithms. Also, evaluate how well the explanatory algorithms can explain the predictions made by the predictive algorithms.
- Highlight what the stakeholder's role in guaranteeing a robust model is and how to convert a data-driven approach to solve a predictive problem into a data-informed approach, where the data patterns and the human knowledge are combined to maximize profit.

1.2.2 Our Published Work

The papers published from the NTL detection case study are the following:

Fraud Detection In Energy Consumption: A supervised Approach

2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) [36] explains our first approach to building an NTL detection system by implementing a supervised classification system. Most of the information from this paper can be read in 3.1 (which includes information that contextualizes the NTL detection problem, including the related work) and in 3.3 (which describes the classification approach), although the two chapters feature information not included in those publications.

A Quality Control Method for Fraud Detection on Utility Customers without an Active Contract

Proceedings of the 33rd Annual ACM Symposium on Applied Computing [31] is our first approach of using an explanatory algorithm to provide explainability in our system that

uses a black-box algorithm. Different aspects of this work are reflected in Chapter 5 that explains the explanatory methods tested in our system (including the LIME approach).

Bridging the Gap between Energy Consumption and Distribution through Non-Technical Loss Detection *Energies* [32]. Analyses business-related aspects as well as other data-related problems from our NTL approach. Different aspects of this work can be read in Chapters 3.4 and 3.5.

Non-Technical Losses Detection in Energy Consumption Focusing on Energy Recovery and Explainability *Journal Track from 2021 IEEE International Conference of Data Science and Advanced Analytics, published in Machine Learning Journal* [34] fully introduces the Shapley values in our system. As we explain in this work, the use of Shapley values made us understand well the shortcomings and problems of our system, and helped us to correct them by implementing several solutions (e.g. by using a regression approach instead of the classification approach until now). This work is explained in Chapter 4.

Explainability in an Industrial Case: Predicting Non-technical Losses in Energy Consumption [30] corresponds to Chapter 5, where we provide an extensive vision of our effort to understand our predictions, ranging from using statistical methods to the use of Shapley values.

A Human-in-the-Loop Approach based on Explainability to Improve NTL Detection *International Workshop on Intelligence-Augmented Anomaly Analytics (ICDM 2021 Workshop)* [33] introduces the possibilities that the combination of human knowledge and explainability offers to the system. A more extensive explanation of this paper is seen in Chapter 6.

Knowledge-Based Segmentation to Improve Accuracy and Explainability in Non-Technical Losses Detection *Workshop on Arti-*

ficial Intelligence in Power and Energy Systems (AIPES), ECAI 2020 [25] analyses the benefits of building specific campaigns for each type of NTL.

From our collaboration with the University of Padova, we have the following publication:

Explainable Predictive Process Monitoring *International Conference on Process Mining* [58] describes how we have explained the predictions made by predictive process monitoring. This work is explained in Chapter 7.2.

Regarding the collaboration with the Universitat de Barcelona, we have the following work published:

The Logic behind NGOs' Aid Allocation: a Complex Choice based on Past Decisions [95] analyses if an explained LSTM model can provide a better understanding than the interpretable Regression approach where NGOs implement their projects. This work is explained in Chapter 8.2. This paper is still under review process in a journal.

Chapter 2

Preliminaries

2.1 Supervised Predictive Models

Being X the labelled instances $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where x_i is the feature vector that represents an instance and y_i the value to be predicted, the supervised model aims to learn the function $f, Y = f(X)$ (i.e. to learn patterns so that it can predict for each x_i the corresponding y_i), wherein a binary classification model Y is either 0 or 1 (or $0 \leq Y \leq 1$ if the model provides probabilities), in a multi-class classification is a label, in a regression model the value to predict is continuous (i.e., $Y \in \mathbb{R}$), and in a ranking model the value to predict corresponds to a numeric rank label, also $Y \in \mathbb{R}$.

The process of building a supervised model is complex. In general, it can be divided into three stages: the data extraction and pre-processing, the model building, and the deployment of the predictive system. A short description of each process is given hereunder.

2.1.1 Data Extraction and Pre-Processing

The initial process consists of extracting the data from the data sources and its preparation to serve as input for the supervised model. In general, data extraction and pre-processing require different tasks [59], briefly explained as follows:

Data Extraction, Transformation and Normalization

The initial step consists of extracting the information from the data sources. In general, the data is stored in databases (or other data structures such as event logs) with a structure that does not suit the predictive algorithms and, therefore, needs to be processed to facilitate the pattern extraction. Moreover, it is necessary to normalize the information, that is that all the information that expresses the same information is provided with the same unit. Finally, some algorithms need a data normalization (i.e. representing all the features in a similar or even equal scale or range) to guarantee a good

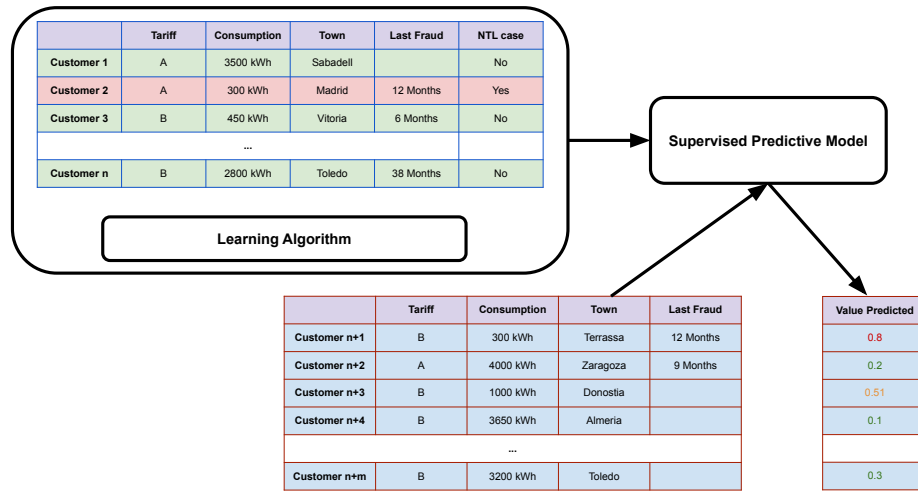


Figure 2.1: Visualization of a supervised model. The learning model has a labelled dataset from which builds the Supervised Model. This model is then used to predict non-visited profiles.

performance. Some of these algorithms that need data normalization are the statistical learning methods (e.g. Linear Regression) or the K-Nearest Neighbors that predicts based on Euclidean distance between instances.

Data Cleaning

The process of correcting the quality of the data when the data is extracted from the original data sources, i.e., to remove incorrect or unnecessary data, detect redundancies and inconsistencies, is referred to as data cleaning. This process includes the decision regarding what to do when data is missing. Depending on the characteristics of the missing data and the problem to be solved, the missing data can be imputed (e.g. through other machine learning algorithms such as the K-Nearest Neighbors [13, 26]) or removed. A similar analysis is done if there exist noisy data in the system (e.g. outliers related to human errors when obtaining the data), where a correction can be applied when possible, otherwise the data can be removed.

Data Reduction

Data Reduction encompasses different techniques that reduce the data dimensionality, maintaining the same essential information and integrity. Removing unnecessary information (e.g. by removing correlated features or applying Principal Component Analysis [123]) helps to avoid multicollinearity in linear models, which leads to numerical instability [5]. In addition, it is also recommended to avoid the curse of dimensionality [136], where large amounts of data both hinder the training efficiency in terms of computational cost but also in obtaining high-quality models. In general,

data reduction does not include reducing the number of instances (as it is generally considered that the more data one has, the better). However, it can be considered beneficial in specific circumstances (e.g. when there is some over-represented type of instance, or when the quality of certain data is in question).

2.1.2 Model Selection, Parameter Tuning and Loss Function

Once the data is prepared, it is necessary to train a model to learn the necessary patterns from the data to make good predictions in unseen instances. In general, the process of building a model follows a recursive feedback structure: a model is built, then its correctness is analyzed (i.e. how well the predictions fits the desired results), to finally update it with improvements and other modifications to building the model again to improve its accuracy.

Loss Function and Optimizers

Building a machine learning model consists of learning patterns from the data, i.e., extracting patterns that reduces the difference between the current's algorithm output and the expected output. This process is iterative, wherein each iteration the model measures how well the system predicts, updating and adjusting the model to reduce the said distance (i.e., the learning process). The function that measures the distance between the expected and current output is the Loss Function.

Some of the most common metrics referred to in this work are the following:

Log Loss Log-Loss measures the performance of a classification models, and it can be defined as follows:

$$LogLoss = -\frac{1}{n} \sum_{i=1}^n [y_i \cdot \log_e(\hat{y}_i) + (1 - y_i) \cdot \log_e(1 - \hat{y}_i)] \quad (2.1)$$

Log Loss error increases as the predicted probability diverges from the label, penalizing confident and wrong predictions.

MAE The Mean Average Error is a regression metric that metrics the average of the difference between the values predicted and the real values, defined as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.2)$$

RMSE The Root Mean Square Error is a regression metric that is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.3)$$

Unlike the MAE, the errors in the RMSE is squared and, therefore, larger errors have more weight in the metric (i.e., it is usually used when larger errors should be avoided). For this reason, this metric is also used in ranking problems (as a point-wise ranking metric).

The process of learning how to classify based on the characteristics of the training dataset is carried out either by algorithm-specific techniques (e.g., the splitting process in the Decision Tree or the euclidean distance in K-Nearest-Neighbors) or by the optimizers. The optimizers are methods used by complex machine learning algorithms (e.g., the algorithms used in this thesis, the Gradient Boosting Decision Tree and deep learning) to minimize the loss function by updating the parameters of the predictive algorithm. There exist different approaches to optimizing a data model, including the Ordinary Least Squares for basic Linear Models (estimated through appropriate linear algebra) or the Newton's Method (e.g., Scoring Algorithm), but the most popular approach in modern machine learning is Gradient Descent and its variants.

The Gradient Descent is a first-order optimization algorithm that calculates the minima of the loss function. Iteratively, the predictive algorithm learns how to update the parameters of the machine learning model (e.g., the weights of a Neural Network model, the support vectors of the Support Vector Machine, or the coefficients in a regression model) in the opposite direction of the gradient of the loss function to the parameters until the loss is minimized. In each iteration, how much the weights are updated is controlled by the learning rate hyperparameter. A very small learning rate will need more time to find a minima, and it also might be easier to find a local minima. In contrast, with a too large learning rate the model would converge to a sub-optimal solution. Therefore, obtaining the optimal learning rate to find the global minima is not trivial and requires a hyperparameter optimization search.

The vanilla Gradient Descent is referred to as Batch Gradient Descent, which uses all the dataset instances to update the parameters. Other approaches are the Stochastic Gradient Descent (that updates the parameters using a single instance) or the mini-batch Gradient Descent (that uses a mini-batch of samples). Other similar optimizers that propose improvements (e.g., by considering an adaptative learning rate) are the Nesterov accelerated gradient, the Adam and the Nadam optimizers [116].

Model Selection

As discussed above, the process of building a predictive algorithm is an optimization problem in which the loss error is minimized. However, the process of selecting an optimal model cannot simply consist of training different models until a global minima is found, since there exist the problem of overfitting. Therefore, it is necessary to simulate the existence of unseen data to control the generalization capability of the trained model. Thus, it is necessary to split the labelled information into two sub-datasets: the training and the validation dataset when comparing different models. With the training dataset we fit the models and evaluate them on the validation dataset that takes on the role of the unseen data. The tuned algorithm that achieves better performance on the validation dataset should be the best candidate in our predictive system. Then, we fit a model using all the labelled information, since we assume that all the samples are i.i.d. (independent and identically distributed) and therefore the conclusions regarding the best model and tuning should not vary when fitting a model with all the labelled information. The results in terms of performance in the actual unseen data (i.e. the test dataset) should be similar to the performance achieved in the validation dataset.

The two most common methods to implement the partition of the labelled information to analyze the generalization performance are the hold-out and the cross-validation methods. The holdout method consists of splitting the information into two sub-datasets, assigning 2/3 to the training and 1/3 to the validation (or similar proportions, 3/4 and 1/4 or 4/5 and 1/5). The cross-validation divides the labelled information into different k-folds of the same size, where each fold is used once as validation dataset and the rest of the folds as training (i.e., its a holdout method repeated k times). K-fold cross validation is preferred over the holdout validation for small datasets to avoid problems related to the split process. Other popular resampling protocols are the Leave-One-Out Cross-Validation (a specific type of k-fold Cross-Validation where each fold has only one instance), the nested Cross-Validation (also referred to as double Cross-Validation, in which two Cross-Validation processes are nested to separate the hyperparameter exploration and the model evaluation to avoid optimistic evaluation and overfitting [139]), the Out-of-Bag error from the ensemble tree models that implements bootstrap aggregating (bagging) [18], and the bootstrap method that implements random sampling with replacement of available labeled information to construct the training dataset, using the labeled information not included in the training dataset as the test dataset [74, 75].

Evaluation Metric

The process of model selection previously explained requires using an evaluation metric to easily compare the different models tested and measure

their performance on the unseen data.

Although the evaluation metric falls under the concept of "how well our model works", its selection is not trivial, as the metric should vary depending on the objective to be achieved with the system. For example, when implementing a medical solution for disease diagnosis using predictive models, the metric must consider that not detecting the disease in a sick patient is a much more serious error than detecting it in a healthy patient. All this is discussed below, where we explain the most common metrics in machine learning discussed in this thesis:

Precision Being the true positives (TP) the relevant instances retrieved (the values can be so much above a threshold, usually 0.5 for classification problems) and the false positive the retrieved non-relevant instances, the precision is defined as follows:

$$precision = \frac{TP}{TP + FP} \quad (2.4)$$

As seen in the formula, the false negatives (i.e., FN) are not considered; therefore, it is indicated when the retrieved instances are taken into account.

Recall Recall corresponds to the fraction of the total amount of relevant instances retrieved (TP) and the total of the relevant instances (i.e., the sum of the TP and FN , the relevant instances not retrieved).

$$recall = \frac{TP}{TP + FN} \quad (2.5)$$

In general, the predictive algorithms that have high precision might have a low recall. Therefore, the use of recall fits the problems in which false positive is preferred over false negative. Returning to the example of the disease detection system, it is preferred to have a high recall and low precision than the other way around. A visual representation of this metric and the precision metric is shown in Figure 2.2.

F-Score The F-score [119] (also known as f1-score or f-measure) is defined as the weighted harmonic mean of the precision and the recall, that is:

$$F = \frac{2 * (precision * recall)}{precision + recall} \quad (2.6)$$

The F-Score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score is equal, while ignoring the true negatives (i.e. the non-retrieved non-relevant instances). There exist weighted versions of the F-Score that prioritize the precision or the recall.

AUC-ROC curve and score A receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate. In Figure 2.3 there is an example of the ROC curve extracted from [36]. The score of this metric corresponds to the proportion of area under the curve (AUC) in that plot, and summarizes in a specific value between 0 and 1 (being 1 the best result) how well the model correctly distinguish the classes in their prediction. This metric, especially the graphical representation, contextualizes better the performance of the predictions, is more open to analysis and interpretation, and has the advantage that it does not consider a threshold to analyze the correctness of the label but analyzes its position in the prediction. However, the scores of this metric are usually over-optimistic when the data to predict is imbalanced [47].

Precision-Recall Curve and Score As analyzed in [41], the Precision-Recall Score is a good alternative to the AUC-ROC when the data is skewed. The Precision-Recall is the weighted mean of precision achieved at each threshold, with the increase in recall from the previous threshold used as the weight.

$$AP = \sum_n (Recall_n - Recall_{n-1}) Precision_n \quad (2.7)$$

Similar to the AUC-ROC curve, the scalar version of this metric scores the area under the curve of the Precision-Recall curve.

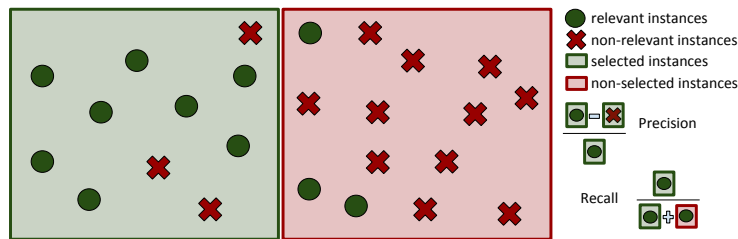


Figure 2.2: In a classification problem, the precision corresponds to the ratio between the selected instances that are relevant and all the selected instances, while recall corresponds to ratio between the relevant selected instances and all the relevant instances.

For the regression model, where the value to predict is a continuous value, the most important metrics are the Mean Average Error and the Root Mean Square Error. These two metrics have already been explained in Section 2.1.2.

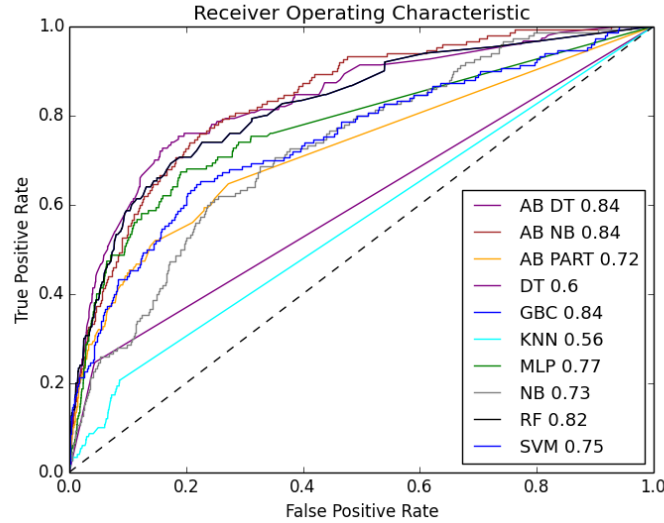


Figure 2.3: AUC-ROC curve from [36], where we explored different algorithms to determine the optimal approach for the NTL detection problem.

For ranking problems, we highlight the following metric:

NDCG : The Normalized Discounted Cumulative Gain ($NCDG_n$) [70] is a measure of ranking quality that evaluates the correctness of our output with a value between 0 and 1 (being 1 the perfect order, and 0 otherwise). This metric allows us a global vision of the correctness of the predictions made without considering one specific threshold.

The $NCDG_n$ is defined as

$$NDCG_n = \frac{DCG_n}{IDCG_n}$$

where, DCG_n is defined as

$$DCG_n = \frac{\sum_{i=1}^n Rel_i - 1}{\log_2(i + 1)}$$

being Rel_i the relevance (i.e., the score in the ranking), and $IDCG_n$, i.e., the *ideal DCG*, corresponds to a perfect ordered DCG for the top n elements of the list.

2.1.3 Deployment of the System and Post-Analysis

Finally, once the optimal approach (i.e. the fitted model) is chosen, a model is built using all available labelled information, and the system predicts the unseen test data. Although it may seem that the process of developing a predictive model is over, there are still elements to analyze, grouped into the following concepts.

Interpretation, Explanation and Visualization

Once the predictions are made, it is necessary to analyze them. By visualizing the predictions or explaining them to the user-end we have a better understanding of what the model has learned and, therefore, it is possible to detect biases or undesired behaviors that have been kept hidden until now. Therefore, this process serves to check the correctness of the model learned and, despite not guaranteeing good results in its deployment, can be helpful to validate it.

Generalization Analysis

Finally, once the predictions are validated, we can metric our model and compare its accuracy with the accuracy from our holdout validation (or cross-validation) tests to analyze if the model generalized correctly.

2.1.4 Challenges in Supervised Predictive Models

The development of machine learning predictive models is not new. The method of processing the data, fitting the data and the post-analysis is well-known to practitioners. However, in Section 1.1.2 we show how, although at first sight the use of data seems to be an extraordinary opportunity in different aspects of our society, achieving successful and fair data science methods can be challenging. The difficulty, in this case, is multifactorial, ranging from the difficulty of the correct implementation of the processes explained in the previous section, the difficulty of understanding the patterns learned by our algorithm, and the validation of the data at source.

Data-Related Challenges: Representing Reality without Bias

The main problem that predictive Data Science models face is the quality of the data. In general, the assumption that the labelled and the unseen instances are i.i.d, i.e. independent and identically distributed, is usually met in data by design. An example of this type of data would be the toy datasets from public repositories that tend to be small and humanly validated. However, when the data comes from observational data produced for other purposes, there is a high probability that the available information does not reliably represent reality, being challenging to guarantee reliability, accuracy, generalizability, or fairness in the predictive model.

In general, the main problem with observational data is selection bias. An example of this is explained in our first case study from Part II that analyzes the development of an NTL system for a utility company. As we analyze in this work, the labelled information was mostly trustful, i.e., the assignation of NTL/non-NTL was done by a professional technician that could be trusted. However, the company decided the visited customers based on their consumption behavior and other business-related consideration. Therefore, most of the company's customers (e.g., customers from

specific regions or customers with a normal consumption curve) were not correctly represented in the labelled dataset. This, together with the fact that the data was non-stationary, implies the existence of dataset-shift [105].

In any case, the development and use of data by design do not guarantee its quality since the designer can introduce human biases in the data. For instance, there is no possible nuance in the label of whether an animal is a dog or a cat, but the definition of creditworthiness of a bank customer, what a good worker is in a company or the value of an elite sports player are values that can be nuanced depending on personal opinions. Moreover, the human role in a data modeling system can prejudice or confirm bias.

All and all, these data-related problems challenges the correct development of a predictive model. The patterns learned might not generalize well and be biased, a fact that can be aggravated if the system learns from its own predictions, reproducing the well-known feedback loop bias [83].

Model Generalization and Evaluation

Assuming that there is no evident bias nor misrepresentation of the case study in the labelled instance, one would expect that the predictive algorithm should be good enough to provide accurate and fair predictions. Nevertheless, in many cases, that is not the case.

Returning to the idea of unbiased data, and also considering that we strive to guarantee a good benchmarking of our system (an element that, as explained in [48] is complicated to guarantee), the analysis of the patterns learned by the model should be the keystone to guaranteeing the correctness, fairness and quality of our model: if the model learns *causal* patterns [100], then our system should be correct. The problem with this approach is that, in many cases, the algorithms learn correlations that have no truly causality in the prediction. An example of this situation is the adversarial examples in machine learning: instances very similar to other instances that the predictive algorithm would properly classify, but with small (and intentional) feature perturbations that trick the machine learning model into making a false prediction. In some cases, the error can be justified (e.g. a spam e-mail that resembles a normal one), but in many cases, the error, from a human point of view, is unjustifiable (see Figure 2.4).

To guarantee the causality of the patterns learned, first, it is necessary to determine what patterns the system has learned. This is a very complicated concept since in many cases, it is not possible to determine what a model has learned. There are (broadly speaking) two different types of predictive algorithms: the interpretable algorithms and the black-box algorithms. The first type of algorithms are considered less accurate and, therefore, scientists tend to use black-box algorithms that can not be interpreted and require explanatory approaches to understand the algorithms. Therefore, if it is very challenging to understand a model, we cannot guarantee that a good model in terms of accuracy is learning good

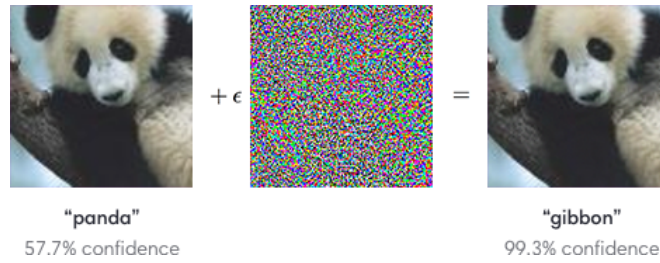


Figure 2.4: Example from [64] where an Artificial Neural Network erroneously classifies a panda as a gibbon after an adversarial modification of the image of the panda. This can be seen as the prove that the deep learning method does not find causal patterns to identify the panda but due to noisy characteristics of the image that are not related to the ground truth

causal patterns or if, on the contrary, it is reproducing unethical biases that prevent the model from generalizing correctly.

2.2 Model Transparency and Explainability

There exist in the Machine Learning community the belief that the trade-off between accuracy and explainability should be considered when using a predictive algorithm: the most accurate algorithms are not interpretable, while the most interpretable algorithms might have limitations in terms of accuracy (Figure 2.5).

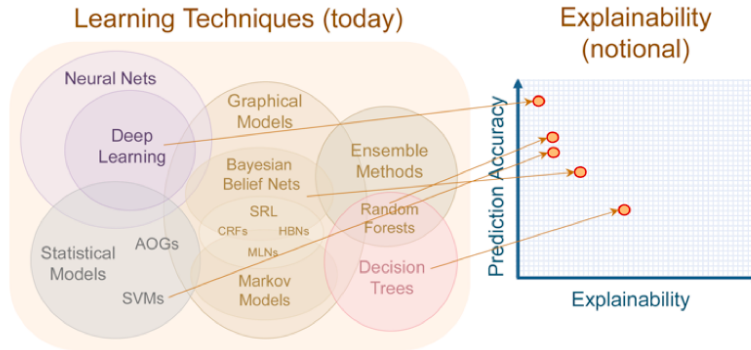


Figure 2.5: Image from [66] that visualize the trade-off between accuracy and explainability and summarize the idea that the more precise the algorithm is, the more difficult it is to explain. This image has been popularized with the DARPA’s XAI program to create tools to understand and trust the most accurate algorithms.

2.2.1 Interpretable Algorithms

Some predictive models are considered interpretable, i.e., one can understand the patterns learned by the model from the training dataset, how these patterns influence the predictions, and even predict how a change in a feature would affect the prediction. The most popular interpretable algorithms are the linear and non-linear Regression models (where the coefficients indicate how each feature influences the output model), the Decision Trees [20] (where one can follow a path in the tree to understand how the model scored an instance), the Decision Rules [69, 29] (where the predictive models are a set of if-else statements that can be easy to understand), the Naive Bayes [89] (where the model is interpretable due to the independence assumption of the features as a conditional probability) or the k-Nearest Neighbors [6, 67] (where the label of the neighbors of an instance explain the prediction of the model).

Below we provide a better explanation of the linear and logistic regression model, and the Decision Tree. Both algorithms appear in the case studies of this thesis.

Linear and Generalized Regression

A linear regression model is a statistical method that models the relationship between a dependent variable (i.e. Y) and the independent variables (i.e. X) as a weighted sum. As indicated by the name, the relation between the variables are linear, defined as follows:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon \quad (2.8)$$

where β corresponds to the coefficients of each independent variable, being β_0 the intercept term (a constant value) and ϵ the error.

Linear Regression is usually used in medicine [63], economics [45] or sociology, fields that quantitative research needs a straightforward interpretation to validate or discard hypothesis; different libraries in R and Python has built-in summary functions that facilitate their interpretation.

The linear regression algorithm finds a correct model if the following assumptions are met in the data:

Linearity: The prediction in a linear model is a weighted linear combination of features. In a way, this is the reason why the linear regression are fully interpretable.

Homoscedasticity: The theory says that in a linear regression model the residuals should equal across the regression line, i.e., the variance of the errors are constant over the feature space. In reality, this is difficult to achieve since the variance of error is higher for higher values to be predicted (e.g. when predicting the price of a house or a car, where there exist a high difference in value between cheap and expensive instances). However, it is intended that the errors are not excessively different proportionally throughout the prediction.

Independence: Observations are independent of each other.

Normality The target outcome should follow a normal distribution. If this assumption is not met, then the confidence intervals of the coefficient weights are not valid.

No Multicollinearity: The independent variables are not highly correlated with each other. The existence of very correlated information hinders the proper assignation of weights in those features.

When these assumptions are not met, the linear models are not a good approach. In some cases, it is possible to bypass these problems by implementing feature engineering (e.g. by removing correlated features). In other cases, it is not possible to fit the problem in a linear regression model as is, e.g., when the value to predict is binary. In these cases, the generalized linear models (GLM) might be an optimal solution. The GLM are an extension of the classical linear models in which the prediction of a linear model is transformed using a nonlinear function to achieve the non-normal output.

The most popular generalized model is the logistic regression that adapts the linear regression to fit the output between 0 and 1, used for classification problems. The logistic function is defined as:

$$P(y^{(i)} = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}))} \quad (2.9)$$

The interpretation for the logistic regression is not as straightforward than the linear regression, since the logistic regression model is a linear model for the log odds:

$$\log\left(\frac{P(y = 1)}{1 - P(y = 1)}\right) = \log\left(\frac{P(y = 1)}{P(y = 0)}\right) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \quad (2.10)$$

Decision Tree

Decision Tree is another well-known interpretable algorithm from machine learning in which the relationship between the features and the value to predict is, in contrast with the regression model, not linear. The Decision Tree is built by recursively splitting the source labelled instances into subsets until the subset can not be partitioned anymore, i.e., if all instances from the set have the same label or if the configuration established determines that it should not be divided anymore. The process of division is used by dividing the set depending on the value of a feature, maximizing the benefit of the split (i.e. reducing the Gain Impurity, Increasing the Information Gain, or reducing the variance of the set).

Another difference between the Decision Tree and the Linear Models is that the Decision Tree offers both instance and modular explanations. At instance level, the interpretation of the Decision Tree consists of following the path the instance goes in the splitting process. In Figure 2.6 there

is an example of a Decision Tree trained with the well-known Iris dataset [53]. To obtain a modular explanation (i.e., the feature importance), it is necessary to compute for each split the benefit (e.g., the increase in terms of Information Gain) after splitting through that feature.

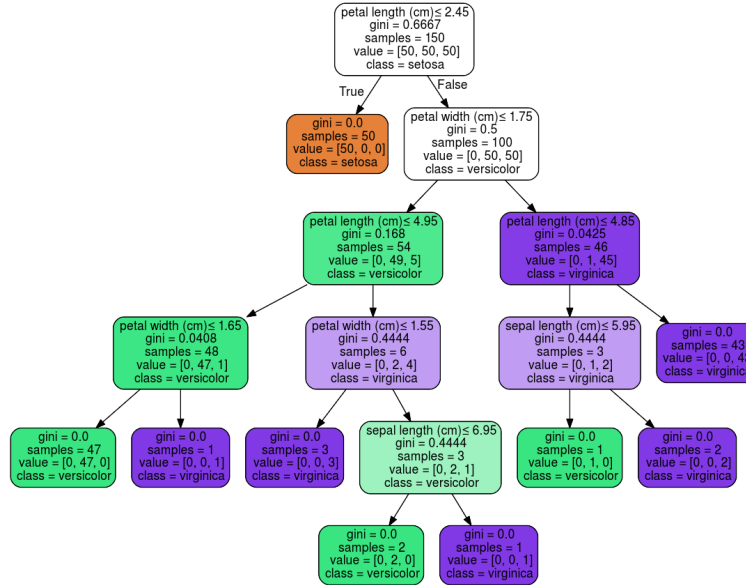


Figure 2.6: Example from Scikit-Learn of a Decision Tree that classify the Iris Dataset. If a Decision Tree is plot, it is easy to follow the feature splitting process to determine the prediction of a non-labelled instance.

The main benefit of using Decision Tree models is its simplicity, but it has several problems. For instance, Decision Tree do not generalize well (i.e. overfits the training dataset) and can be unstable.

2.2.2 Black-Box Algorithms

As seen in previous Section 2.2.1, the interpretable algorithms consist of algorithms that are interpretable due to their simplicity, in many cases thanks to several assumptions that need to be met to guarantee good predictions. In contrast, the black-box algorithms are non-interpretable algorithms. A priori, these algorithms offer better prediction capacity due to their complexity but at the cost of the aforementioned lack of transparency.

The most paradigmatic black-box algorithms are the Gradient Boosting Ensemble Tree algorithm (and all the ensemble methods) and the deep learning algorithms. Both algorithms are used in this thesis.

The Gradient Boosting Ensemble Tree Algorithm

There exist in the literature different approaches to combining shallow trees to increase the performance, reducing bias and variance. The bagging Ensemble Trees (that consist of training a bunch of individual models in

a parallel way, where each model is trained by a random subset of the instances, and predict as an average of each tree [17]) or the Random Forest (that can be defined as a bagging process where, for each tree, only a subset of features are used [19]) are two well-known examples in the literature, used for many different predictive problems.

In [56] there is the explanation of a different ensemble approach called Boosting, i.e., the AdaBoost. This approach combines different tree algorithms as follows:

1. An initial tree model is trained.
2. A new tree model is trained. In this case, the weights of the misclassified instances of the previous tree are increased to focus on their correct prediction. This process is done iteratively N times.
3. The final prediction corresponds to a weighted average of the predictions of each tree, with greater weight given to those trees with higher accuracy.

The Gradient Boosting Tree is a different approach to combining weak learners. In contrast to the AdaBoost approach that builds independent Decision Tree Model and combines them, the Gradient Boosting Ensemble builds a boosted method where the tree at the $n + 1$ stage aims to minimize the errors stochastically from the n stage (see Figure 2.7). That is:

1. An initial Tree is trained to minimize the function loss, e.g. the LogLoss function for a classification model or the Mean Average error for a regression model.
2. A new tree is trained to correct the error of the previous tree, i.e. the residuals. This process is performed iteratively until the residuals are zero or the configuration of its hyperparameters so indicates.
3. The final model corresponds to the prediction done from the first decision tree until the last tree, i.e. there is no combination of different predictions but one unique prediction.

The boosting of trees has low interpretability: for the first (or few first tree) it is possible to follow the splitting process of the trees, but in general the ensemble has several hundred or thousand trees that make manual interpretation unfeasible.

Some state-of-the-art implementations in python are the XGBoost [28], the LightGBM [73] and CatBoost [103], methods that introduce several improvements (e.g. better regularization to avoid overfitting) to the classical Gradient Boosting approach.

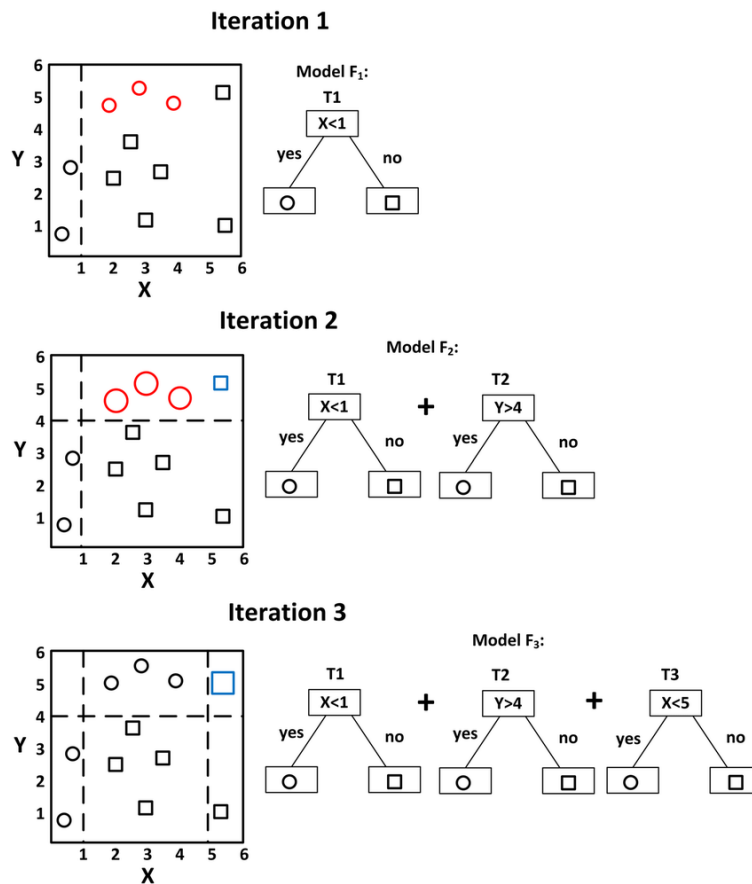


Figure 2.7: Example extracted from [145] to explain the Gradient Boosting Tree model: In each iteration, the new tree n reduce the errors of the model.

The Deep Learning Algorithm

Artificial neural networks (ANN) (Figure 2.8) is a predictive model consisting of consecutive layers of neurons, i.e., computing units, in which each neuron receives different weighted input values and produces an output. In the classical Neural Network (with forward propagation), the input layer receives the input instances, the layers (and their neurons) process the instances, and finally predict the output in the last layer. The error is then back-propagated and the weights of the neurons are updated (via the optimizer) to reduce the error done by the predictive model. There exist different methods of connecting the layers in a neural networks, and also different activations of the neurons (i.e. different ways of generating an output with the same input). Initially, artificial neural network were shallow, i.e., had only one or very few layers. Over the last 20 years, in conjunction with the increase in computer power, we have seen artificial neural network become deeper and more complex: these more complex models correspond to deep learning.

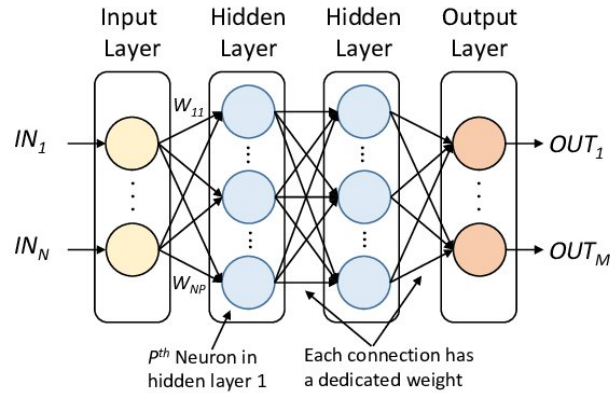


Figure 2.8: Image from [68]: there is an input layer, an output layer, and different hidden layers.

The classical neural network is the feedforward backpropagated neural network, where the data only moves from input to output without loops and cycles. The recurrent neural network is an evolution of the feedforward neural network, where loops and cycles are included to maintain historical information. It is used specially in image and speech processing, but has the problem of gradient vanishing and exploding: in recurrent neural networks (or in very deep learning models), there is the stability issue where in certain scenarios (e.g., a neuron with a large input and a sigmoid activation of a deep neural network learned using backpropagation) the derivative becomes too small to learn (gradient vanishing) or too large (gradient exploding). This is solve by using other activation functions (e.g. Relu) or other deep learning structures (e.g. the LSTM that introduces memory units to store information in an undefined time with no degradation).

The deep learning algorithms are considered the less interpretable predictive algorithms.

2.3 Explainable AI

Explainable AI (XAI) is a new term popularized by DARPA’s XAI program [66] that refers to techniques, methods and algorithms that explain the predictions made by the black-box algorithms and arises as a response to the technical, social and legal concerns of the increasing use of artificial intelligence techniques (especially these black-box algorithms) in our lives, as we introduced in the Introduction (Section 1.1).

Explaining a prediction is a general term that encompasses different techniques and approaches depending on the objective to be achieved (i.e., the reason why we want to understand our algorithm) but also the algorithm to be explained, and can include approaches that present textual, numerical or visual information that allows the human to understand the predictions.

2.3.1 XAI goals

The XAI community has highlighted many different goals around the need to explain an algorithm. Nevertheless, all of them can be categorized as either a technical goal (i.e., whether the system is trusted) or an ethic goal (i.e., whether the model is unfair).

How Reliable are the Predictions? Technical Analysis In [110], trust of a prediction is defined as *“whether a user trusts an individual prediction sufficiently to take some action based on it”* and model trust as *“whether the user trusts a model to behave in reasonable ways if deployed”*. Both definitions of trustworthiness are intrinsically related to the patterns learned by the model: if the model learns causal patterns, then it can be trusted, since it correctly infers the causal relationship between the features and the target. However, as broadly analyzed in the literature [99, 100], the predictive algorithms can not understand the quality of the patterns learned (i.e., whether the pattern corresponds to a causal relationship, to a logical but non-causal correlation, or whether the pattern corresponds to a mere coincidence that should be avoided) and therefore a human validation of the model with high knowledge of the automated process is necessary. For this reason, much literature emphasizes that XAI must also guarantee model transparency, i.e., the model must be highly informative of the learned patterns to facilitate the detection of undesired behaviors of the predictive algorithms. Moreover, this transparency should not only focus on the validation of the trained model, but also on the acquisition of new knowledge (e.g. acknowledgment of trustful patterns previously unknown) and also facilitate the implementation of different machine learning techniques (e.g. transfer learning).

Once a reliable and transparent model that systematically learns robust and reliable patterns is achieved, the model’s users will have confidence in its use. Confidence in the model is key to relying on data to make decisions in the industry (often referred to as data-driven solutions).

How Fair is the Model? Legal and Ethic Concerns From a social perspective, explainable AI is the solution to control and audit machine learning methods that automatize processes in which citizens are affected. The aim of this control is to detect biases and other data problems that would reproduce and entrench historical prejudices against citizen communities, and thus ensure fairness in automated predictions. To ensure this control by regulatory institutions or even the ordinary citizen, it is necessary to guarantee the accessibility of these algorithms, i.e., that non-technical users affected by these decisions should be able to access the prediction made by an algorithm and the patterns learned.

2.3.2 XAI Approaches

Explaining a predictive black-box model is dependent on human judgment and is therefore difficult to uniquely mathematize. There are two operational definitions of explainability: model-level and instance-level.

Model-Level Explainability If M is a trained predictive model that receives instances $x = (v_1, \dots, v_n)$ to predict, a model-level or global explanation of M is a vector (w_1, \dots, w_n) that describes how each feature x_i globally influences the predictions made by M , typically computed on the training instances used to build M .

Instance Level Explainability An instance-level or local prediction provides such a vector for a specific instance x , therefore how M is influenced by each feature to produce its specific prediction $M(x)$.

In the following we explain the explanatory methods used in this thesis, as well as a brief summary of other explanatory approaches in the literature.

Feature Importance

In tree models (i.e., a decision tree, or an ensemble of trees) it is common to analyse the importance of each feature by computing the Feature Importance, providing a model-level explanation. Depending on how it is computed, the importance of a feature can be divided into *prediction* and *occurrence* methods:

- Prediction methods: they analyze the influence of feature values in the predictions made by the model. This naive definition includes, for instance, the Random Forest from Scikit-learn [101] (that evaluates the Gini impurity of the samples of the nodes decrease after a split using that feature), or the *LossFunctionChange* from Catboost [103] that evaluates how the prediction changes if that feature is removed.
- Occurrence methods: Measure the importance of the feature by analyzing its occurrences in the training process, i.e. how many times the feature has been used in the splitting process, usually referred to as *weight* or *frequency*, or the number of instances in the node split by that feature, usually referred to as *coverage*.

Local Surrogate Models

Local surrogate models are simple interpretable models that aim to replicate the prediction made by complex black-box models for one specific prediction, i.e. provide an instance-level explanation: Let M be a predictive model that the surrogate model aims to explain, x be the instance to explain, and L_n be an interpretable model (e.g., a linear Regression) trained

on n instances chosen somehow, then we would like to have $L_n(x) \simeq M(x)$ while keeping the model complexity of L_n as low as possible e.g., using as few features as possible to provide a simple explanation. Different methods differ in the type of model L_n and the instances used to build it, which may be selected from the training set or generated synthetically.

A state-of-the-art approach to local surrogate models is LIME (Local interpretable model-agnostic explanation) [111]. The idea of LIME is to analyze how the prediction of the interpretable algorithm changes based on the absence-presence of a feature. For image classification, the algorithm creates superpixels (i.e. portion of the image), and analyze how the prediction changes based on the absence-presence of this superpixel in the image. For text classification, words are included-removed from the text to understand how influence in the output. For classical tabular data, LIME perturbs each feature of x independently, using a normal distribution with the same mean and standard deviation.

Shapley Values

Originally, Shapley values [121] were conceptualized as a game theory approach to computing the fair distribution of payout among players in a cooperative game. The SHAP library adapts this idea to provide explanations for predictive models: being the features from an instance the players and the difference between the prediction made by the predictive model and the average prediction (i.e., the base value) the payout, it can be adapted to understand the role of each feature in the prediction process. This approach provides both a global and local explanation: the global explanation is the sum of the local explanations and, therefore, both explanations are consistent and have a common foundation.

The Shapley values of a feature value in instance x are usually defined as follows:

$$\phi_j(val) = \sum_{S \subseteq \{v_{i1}, v_{i2}, \dots, v_{im}\} \setminus \{v_j\}} \frac{|S|!(p - |S| - 1)!}{p!} (val(S \cup \{v_j\}) - val(S)) \quad (2.11)$$

where p corresponds to the number of features, S to a subset of the instance features, and val corresponds to the function that indicates the payout for that features. In the equation, the difference between the val corresponds to the marginal value of adding the feature in the prediction for a particular subset of features S . The summand denotes all the possible subsets S that can be done without including the feature from which the Shapley values are calculated, i.e., v_j . Finally, $\frac{|S|!(p - |S| - 1)!}{p!}$ corresponds to the permutations that can be made with subset size $|S|$, in order to properly distribute the marginal values among all the features of the instance. All possible subsets of features are considered, and the effect on the prediction of including the feature to each subset is observed.

SHAP library offers different *core explainers* (i.e., methods to compute the Shapley values). In this thesis we use two methods: the Tree SHAP (to compute the Shapley values for tree models) used in Part II and Deep SHAP (for deep learning models, that approximates the Shapley values through an enhanced version of the DeepLIFT [125] algorithm) used in Part III.

Other Approaches

In this thesis we highlight Feature Importance, Local Surrogate Models (i.e., LIME), and Shapley values, as these are the approaches used in our case studies (with tabular data); we provide an in-depth analysis of each approach for tabular data in section 5. But there are many different approaches, which we summarize now.

One of the simplest approaches to understanding how a feature influences an outcome is through the partial dependence plot [57, 65], which consists of showing through a plot the marginal effect (e.g., a monotonic relationship) of the value of the feature on the predicted outcome. The simplicity of this approach implies certain advantages (e.g., it is easy to implement, intuitive and straightforward) but also several disadvantages (e.g., it omits the feature distribution of the variables, and also assumes that the features are independent).

Another rather simple model-agnostic approach is counterfactual explanations [138, 39]. A counterfactual explanation aims to determine which smallest change in the values of an instance’s features changes its prediction. This approach provides a simple approach to obtain good explanations. However, it is possible to find different explanations for the same instance, so it is challenging to determine which one should be considered the correct one. This approach shares many similarities with the adversarial examples approach [131], with the difference that the former aims to explain the model and the latter aims to deceive it (as exemplified in Figure 2.4).

Anchors [112] is another approach that finds which feature values fix the prediction. In other words, this approach provides a decision rule explanation (i.e., very simple to understand “if-else” explanations) to explain the predictions of a model. This approach shares many similarities with LIME ¹, building their explanations with a perturbation-based strategy.

Finally, we would like to focus briefly on the existing approaches to provide explainability to deep learning models for non-tabular data (e.g., image and text classification problems). Both LIME and SHAP (which in turn provides two different explainers, the one based on the aforementioned DeepLIFT algorithm, and one based on the Integrated Gradients[130]) are also popular solutions in deep learning for image and text classifica-

¹ Both approaches have the same authors.

tion. Broadly speaking, the existing solutions are either model-agnostic approaches that base their explanation on the perturbation of the data (e.g., analyzing the difference in prediction after pixel occlusion or perturbation), or they are gradient-based solutions (which are specific to deep learning models) that explain the model through the gradients of the training process.

2.4 Discussion

The development of a predictive algorithm, as we have explained in this chapter, is not trivial. It requires obtaining and processing data from different sources, exploiting the data knowledge with the optimally tuned predictive algorithm, in order to achieve the desired predictions. This whole process, even if done with utmost care, often fails to achieve the desired results, as there are several problems (e.g. the reliability and fairness of the patterns learned) that are difficult to analyze and correct if they are not optimal. That said, the data science community is aware of this, and they propose different methods that allow us to better understand what our algorithm is learning, allowing stakeholders and data scientists to correct these problems to obtain a better predictive system. In this thesis we analyze the application of predictive models in 3 different fields, with the aim of achieving predictions that are both accurate and highly transparent and useful for the researcher or the company using it.

Part II

**Human-Aware NTL
Detection**

Our initial application area analyzes the development of a Non-Technical Losses Detection system for a utility company developed from 2013 to 2020.

Hereunder we detail the evolution of this NTL detection system, from a black-box algorithm to a fully explained system, step by step, providing information of different aspects of our research such as the detection of biases, the lack of robustness, as well as the optimal explanatory approach between the most recent state-of-the-art techniques seen in the literature.

As far as the authors are aware, this work represents the first piece of research that tackles the problems in interpretability of the supervised Non-Technical Losses detection problem (as well as the problems that arise from this such as lack of robustness or dataset-shift). Similarly, it is one of the very first pieces of work done to explore the possibilities that explainability provide in the implementation of predictive data science in the industry.

Papers

Fraud Detection In Energy Consumption: A supervised Approach *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* [36].

A Quality Control Method for Fraud Detection on Utility Customers without an Active Contract *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* [31].

Bridging the Gap between Energy Consumption and Distribution through Non-Technical Loss Detection *Energies* [32].

Explainability in an Industrial Case: Predicting Non-technical Losses in Energy Consumption [30].

Non-Technical Losses Detection in Energy Consumption Focusing on Energy Recovery and Explainability *Journal Track from 2021 IEEE International Conference of Data Science and Advanced Analytics, published in Machine Learning Journal* [34].

Knowledge-Based Segmentation to Improve Accuracy and Explainability in Non-Technical Losses Detection *Workshop on Artificial Intelligence in Power and Energy Systems (AIPES), ECAI 2020* [25].

Chapter 3

Preliminaries in Non-Technical Losses Detection

3.1 Context of the Application Area

Utility companies provide an essential service to developed societies, supplying electricity, gas and water to homes, businesses and factories. The infrastructure necessary to guarantee services ranges from the kilometers of pipes or lines that transport the energy to the millions of meters that monitor the consumption of individual customers. An important problem that these companies face is the imbalance between the energy billed with respect to the energy provided, called *energy losses*. *Non-technical losses* is a widely used, somewhat euphemistic name including fraud and meter malfunctions among others.

Methods for committing fraud include splicing the pipes to bypass the meter, tampering with the meter to stop it or to slow it down, and simply connecting to the distribution network without even having a contract with the company or a meter. On the other hand, an accidental malfunction of the meter also results in net energy loss for the company. Both are issues that the company wants to detect and fix as soon as possible, and the detection schemes are essentially the same.

Most cases of NTL involving meter tampering or malfunctioning can be detected by direct inspection by a trained technician. However, it is extremely expensive to send technicians to inspect a large number of meters. Therefore, companies usually perform a pre-selection of a subset of meters to be directly verified by technicians in a given period of time and area, a concept that we call a *campaign*. Every customer visit has a cost, so in order to be worthwhile campaigns need to have relatively high precision (i.e., percentage of problems detected with respect to the number of meters verified). Company gains, on the other hand, are directly proportionally to campaign recall (i.e., fraction of the existing fraud that is detected), so

campaign design is all about the classical precision-recall trade-off.

Traditionally, campaigns are based on simple sets of rules indicating fraud (e.g., abrupt decrease of consumption, or no consumption during a long period of time). These rules can be used to detect the fraudulent/irregular customers, but achieve a low success rate, not much higher than selecting customers for the campaign at random. This can be explained by many other reasons besides fraud (e.g., a customer spends a long convalescence in hospital, or the house is a second residence that does not follow the consumption patterns of an all-year primary home). With the booming of artificial intelligence in industries, the utility companies have seen the possibility of exploiting the large amount of data available from their customers, especially since the installation of smart electricity meters.

Naturgy is a utility company distributing both electricity and gas in Spain and 26 countries on 5 continents. Their classical approach of detecting NTL was based on simple rules with only fair results. In mid 2013, the company approached UPC researchers looking to improve the fraud and anomaly detection rate of their current campaigns. The first dataset received from the company was to be used to generate a static campaign (no software system). It came from a medium-size city with a few thousand electricity customers and another few thousand gas customers, and contained contract information (e.g., the tariff or the age of the meter) and about two years of consumption information.

After this initial dataset, we received four more datasets with similar characteristics (i.e., cities with thousands of customers), either of gas or electricity or both. From the sixth dataset onwards, and in view of the good results of the initial campaigns, we were given access to data on a national scale, with information of several million customers in gas and another several million electricity customers; around a million customers are customers of both utilities; the project gradually evolved from the original goal of “mining” one specific dataset to create a one-off campaign to the development of a software system that connects to the company’s operational system to generate both routine campaigns and on-demand parameterized campaigns, implementing data mining techniques to manage feedback, and investigate the usefulness of new features, among other functionalities.

3.2 Related Work

There exist in the literature several techniques to detect NTL based on customer data. A very common approach to detect NTL is to implement a supervised system. In [23], a similar approach to ours was presented (it uses Gradient Boosting models and is implemented in Spain); in [90] an approach that uses Support Vector Machines is reported, and in [91] an update of the previous work in which the addition of Fuzzy Rules improves the detection system; in [37, 102, 55, 52, 142] five examples of using neural networks to detect NTL are described, and in [107, 108] two interesting

approaches that differ slightly from the classical supervised algorithms that use the Optimal-Path Forest Classifier [96].

In contrast to the aforementioned supervised techniques, there are also other different unsupervised approaches to detect NTL in the literature. The typical technique is the clustering method, seen in [12] or [10], but we can see other approaches such as [24] that uses unsupervised neural networks (Self-Organizing Maps). A different technique is seen in [127] or [80], which are two examples of using statistical process control method in the detection of anomalies from a more industrial process control point of view.

Despite the fact that the more classical machine learning approaches use supervised and unsupervised methods, other alternatives exist in the literature. For instance, in [27] there is an example of an expert system; also, [71] presents an approach for analyzing the load flow, and in [143] a method based on the sensors of the system is proposed.

In [62] there is a technical survey that analyses the challenges seen in different papers such as the dataset shift, the features built and scalability, and in [86] there is a more classical survey that summarizes the approaches seen in the literature.

3.3 NTL Detection: Baseline Approach

Our work published in [36] explains our Non-Technical Losses Detection system based on a classification model at the time of the publication of the paper. We use that paper as a reference to explain our first approach in this case study, and update the information when necessary with the updates introduced in [32].

3.3.1 Data Processing

The sources of data used to build our NTL detection system include consumption and profile data, historical fraud cases, and some external information.

Consumption data This is the data reflecting the energy used by the customers. It includes meter readings as well as billing statements - the invoices the company charges to the customer based on the meter readings or, when not available, an estimate by the company based on historical information¹. The *consumption reading* corresponding to the difference between consecutive meter readings of the customer is our main source to extract the consumption data; Figure 3.3 is an example. Note that from this information one cannot detect the fraud cases in which someone

¹ This is true before the installation of smart meters of around 2016 onwards

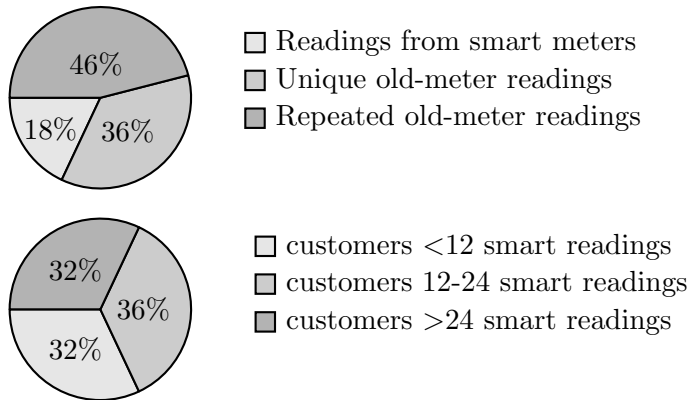


Figure 3.1: Information from the quality of the information available in [36]: Half the customers for electricity had smart meters. From all the readings we had, 18% came from smart meters, 36% were unique readings from old meters and 46% were repeated. Of the customers that had smart meters, around 32% had less than 12 smart-meter readings, 36% had between 12 and 24 readings and the rest had more than 24 readings.

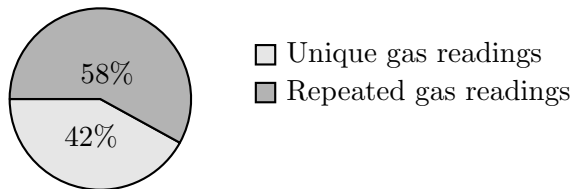



Figure 3.2: 42% of the readings from gas are unique, and 58% are repeated. In gas, the absence of smart meters makes the information available less reliable.

connects directly to the grid rather than e.g., manipulating the meter; this type of fraud can only be detected further upstream.

The origin and reliability of consumption data for our first approach was varied. In electricity, about half the customers had smart meters that send customer consumption to the utility company monthly and reliably. The other customers had old meters that required manual readings: customers were expected to use any one of a number of options to send in the reading (calling the company, sending the reading via mail or a mobile app, or writing it down on a sheet or form available in the building). In gas there were (and there are still) no smart meters, so all the information is sent manually. When there were no smart meters and the customer had not provided the reading, the company needed to estimate the consumption of the customer using reference values from similar historical periods. Customer-generated readings were notoriously unreliable and error-prone, and many customers simply did not send any for many months in a row. Eventually a technician would be sent to read the meter, but only after sev-



Month	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Old meter readings	125	125	140	400	182	182	0	182	230	230	255	255	-	-	-	-	-	-
Smart meter readings	-	-	-	-	-	-	-	-	-	-	-	-	285	295	310	340	355	370
Calculation	$(140-120)/2$		$(182-140)/2$		$(230-182)/4$				$(255-230)/2$		$(285-255)/2$		$\text{Consumption}_{\text{month } n+1} - \text{Consumption}_{\text{month } n}$					
Consumption	10	10	21	21	12	12	12	12	12.5	12.5	15	15	10	15	30	15	15	

Figure 3.3: Example customer consumption record. In light gray, useful information (new readings and monthly consumption). In dark gray, non-useful information (i.e., erroneous information is crossed out, in italics repeated readings, with a dash months without readings). Non-smart meters give less accurate monthly consumptions; in this case, the first October reading is obviously wrong, so apportioned consumption because we have new readings every two months. Since the second July, consumption readings are reliable because a smart meter was installed.

eral months of estimated readings. Fortunately, each consumption record was labelled as company verified, customer provided or estimated, so we could assign them different reliabilities.

Erroneous, absurd or missing readings for non-smart meters, the co-existence of two metering systems with different reading periodicity (1 vs. 2 months between readings), and the fact that some customers changed from one system to another in the process (a small proportion of clients had more than two years of smart-readings in 2016) were complications that we had to deal with when reading, parsing and standardizing consumption data.

Static Profile Data Within this concept we include information related to the customer’s contract with the company (e.g., the tariff), customer information (e.g., their address) and characteristics of the equipment (e.g., the age and model of the meter, whether it is inside or outside the house). This information can be used to categorize the customer; for example, the tariff indicates whether it is a home, a shop or a restaurant, and whether a gas cooker or central heating is present.

Historical Fraud Cases For the first campaign, we were also provided with a list of customers who had committed fraud and were discovered in baseline campaigns carried out by the company in the last few years. The list included several tens of thousands of verified fraud cases which could be merged with their consumption data to investigate fraud patterns. No “negative cases” (i.e., clients that were known to not commit fraud) were received.

FEATURE	DEFINITION
Abrupt decrease in consumption	A reduction of $x\%$ in consumption during n months in comparison to the previous n months.
Abrupt decrease in equivalent consumption	A reduction of $x\%$ in consumption during n months in comparison to the same months from the previous year.
Long period of low consumption	A reduction of $x\%$ in consumption during n months in comparison to the average.
Consumption discrepancy	High/medium/low discrepancy between a consumption in comparison to the average.
Decrease in consumption (correlation)	A consumption reduction during n months using Pearson correlation.
Consumption peak	Consumption in a month $x\%$ times higher/lower than the previous and the following month.
Billing/consumption similarity	Similarity between the consumption computed and the billing (only in gas).
Unknown consumption	Number of consecutive estimated readings, consecutive 0 consumption (in the present and historically),
Difference between years	Highest difference between the consumption of two consecutive years, or if the customer has been consuming less year after year (only in gas).
Gas consumption without electricity	Gas consumption without electricity.
Difference of consumption	High/medium/low difference between the higher and the lower consumption of the customer.
Ratio difference of consumption	High/medium/low difference between the average consumption of the customer and the average consumption.
Negative Consumption	Whether the customer has a negative consumption (e.g., count was reset at the installation of a new meter).
Reading correction	Whether the consumption obtained required a correction.

Table 3.1: Types of features extracted from the data available. In some cases, several versions of the feature are included in the system (e.g., abrupt decrease in consumption, where different $x\%$ and n lead to different features, or using Spanish and regional averages to compute different features).

External Information We also used the Koppen climate classification data [4, 3] of the different regions under study, as climate obviously affects energy consumption patterns, as well as census data about the socioeconomic classification of cities and regions.

3.3.2 Creating a Classification Problem

Below there are the three main issues we had to tackle when transforming the data into a classification problem.

Unreliability in Consumption Data As discussed before, consumption data as given by the readings was not an exact picture of the customer consumption but an approximation, because of the heterogeneous channels used to obtain the readings. For instance, in Figure 3.3 it can be seen that the reading in February is wrong, according to the readings in January and March. A small number of heuristics were designed to correct or discard

FEATURE	DEFINITION
Tariff	Tariff of the customer.
Location of the meter	Location of the meter (e.g., inside/outside the house)
Contracted power	Contracted power (only in electricity).
Electric tension	Electric tension (only in electricity).
Abnormal contractual status	The client has abnormal contractual status (e.g., has canceled the contract with the company).
Regional income	Whether the customer lives in a region with an average income above, similar or below the Spanish average.
Climatology	Köppen climatology classification of the region where the customer lives.
Reading periodicity	Reading periodicity (1 or 2 months, only in electricity).
Number of readings	High/medium/low number of different readings from the customer.
Province	The province where the customer lives.
Capital province	Whether the customer lives in a province capital.
Date information	Age of the meter, date of installation and contract.
Smart meters	Whether the customer has a smart meter.
Old fraud	Whether the customer was detected as fraudulent by the company in gas or electricity in the past.

Table 3.2: Types of features extracted from the data available. In some cases, several versions of the feature are included in the system (e.g. abrupt decrease in consumption, where different $x\%$ and n lead to different features, or using Spanish and regional averages to compute different features).

suspicious or inconsistent data, both in gas and electricity (e.g., discard a reading smaller than the previous and the following reading, or discard an absurdly high reading for a month, replacing them with interpolations). Billing information was also used, but as a secondary source compared to actual or estimated readings.

Statistical Evaluation of the Features The main metric used to evaluate the features was the odds ratio (the odds that an outcome will occur given a particular case, in comparison to the odds that the outcome will occur otherwise), from now on denoted as *OR*. For each feature, we analyzed the OR between the results from the feedback (e.g., the OR between fraudulent and non-fraudulent customers), denoted as *ORPN*, as well as the OR between the fraudulent clients against all the clients not included in any campaign, denoted as *ORPG*. Table 3.5 shows the odds ratios of some features used.

Feature Construction Consumption, profile, and external information were combined to create a number of features. These features are the result of an evolution of the initial rules used by the company for their baseline campaigns (e.g., the detection of customers with a long period of time with no consumption). Yet, each rule used in the baseline campaigns used one or at most two of the features, while we expected machine-learning-based algorithms to be more accurate by taking into account hundreds of features as well as their combinations.

Some features focused on the behavior of the customers in comparison to themselves: an abrupt or gradual decrease in their consumption exemplified in Figure 3.4, a repeated lack of reported readings, a substantially different pattern from previous years, consumption peaks (as seen in Figure 3.5), the difference between the minimum and the maximum consumption of the customer, etc.

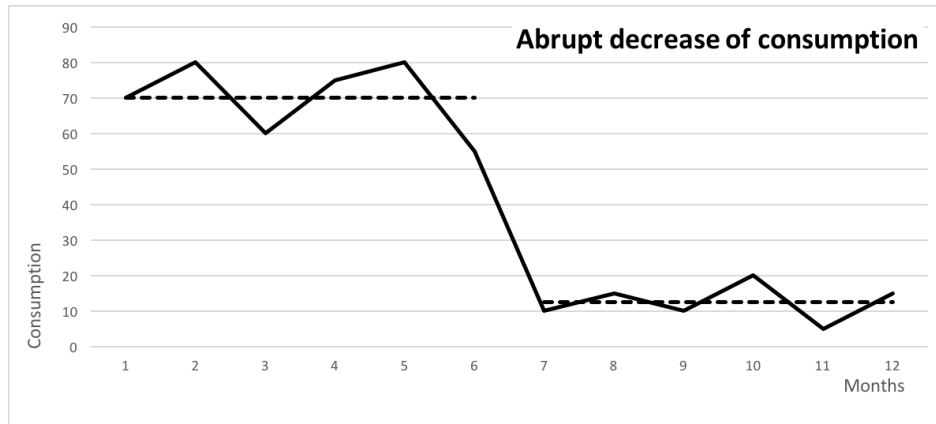


Figure 3.4: Example variable: consumption drop.

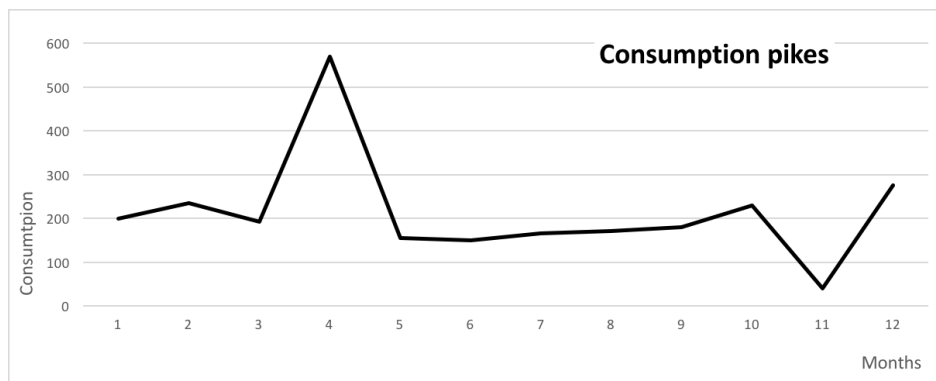


Figure 3.5: Example variable: consumption peaks. We can see that the customer has both a positive peak (4th month) and a negative peak (11th consumption) in its consumption curve.

Other variables measured the inconsistencies of customer consumption in comparison to other similar customers. For example, long periods of time where the customer consumes much less than the average of the customers with the same tariff (exemplified in Figure 3.6), high discrepancy between the consumption curve of the customer and the typical consumption curve of the customers with the same tariff (a flat consumption line showing no seasonality pattern may indicate that the metering has been tampered with so as not to exceed a certain metering speed, as seen in 3.7), nominal difference between the consumption of the customer and the

average consumption of the clients with the same tariff, etc.

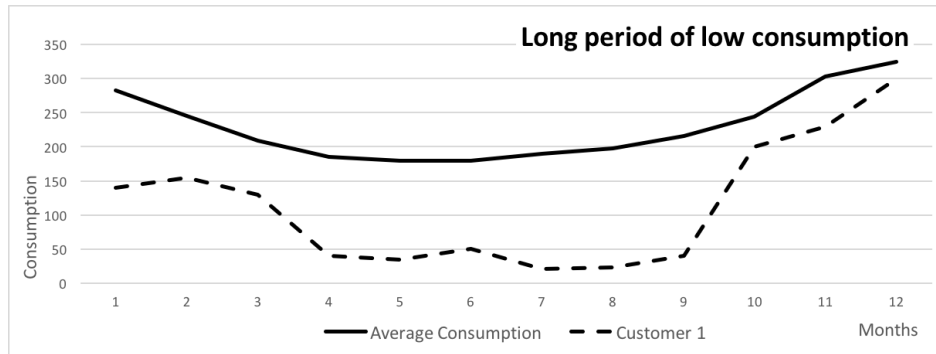


Figure 3.6: Example variable: long period of low consumption. As we can see, we have a period of time (from months 4 to 9) where the customer consumes much less than the average.

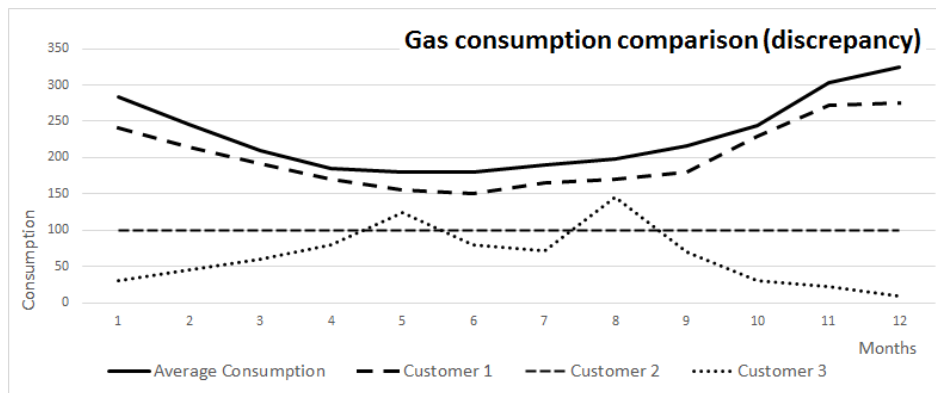


Figure 3.7: Example variable: consumption discrepancy. Customer 1 has a similar consumption curve to the average consumption. On the other hand, both customers 2 and 3 have an abnormal curve that might be an indicator of fraud.

It is worth mentioning the features that combined information from gas and electricity consumption (e.g. the behaviour comparison between the gas and electricity consumption, or the consumption of gas without electricity consumption, exemplified in 3.8). These features, tested statistically, were only included in the most recent campaigns, which are the more successful ones.

From the static data we also extracted some features; the province where the customer lives as well as its climatology, the location of the meter, the date of installation of the meter, etc. Table 3.2 contains a list with the feature types included in our system.

These variables were all binarized, for uniformity and simplicity; in particular, this avoided problems with algorithms that are too sensitive to

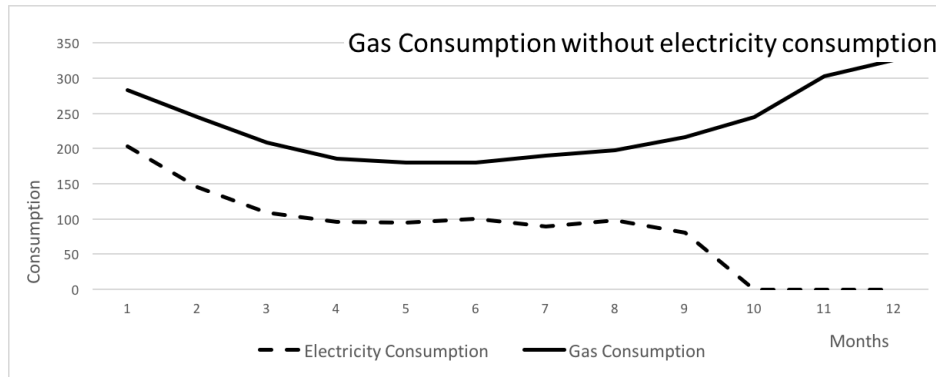


Figure 3.8: Example variable: Gas Consumption without electricity consumption. Gas heater needs electricity to work.

outliers or extreme values. For example, one binary variable was introduced for each tariff type that a customer may have. In some cases, several variants of the same variable were introduced, corresponding to different horizons or thresholds; for example, whether there has been a reduction of $x\%$ in the last n months generates many variables for varying x and n . For each candidate variable, the ORPG was checked, and those variables with values near 1 that were not useful (e.g., did not have profiling information) were removed.

All in all, the number of variables included in our first approach reached 250 features in the electricity campaign and 150 features in the gas campaign. At the end of this process, a customer is represented (with the information available at the time of generating a campaign) as a vector of binary variables, which we call the *customer profile*.

Imbalanced Classification Problem We created a classification problem by labelling each customer profile in a potential training dataset either with the positive class (P), representing fraudulent behaviors, or the negative class (N) representing non-fraudulent behavior. For populating P, we considered customers from the historical fraud cases and those detected as fraud in previous campaigns. Populating N was a problem in the initial campaign, as we did not have certified non-fraud cases; we simply took a random sample of all customers, which should be approximately correct under the assumption that fraud prevalence is low enough. As we started receiving feedback from the first campaigns, we did have certified negatives.

The prediction desired from the system could be a P/N value. In this case the campaign is simply an unordered set of suspicious customers (predicted to be P). The company, however, preferred to have a *fraud scoring*, or probability of being fraud, for each customer, which makes the campaign an ordered list; this allowed us to detect in-place that a campaign had entered a point of diminishing returns.

3.3.3 The Process

Our NTL detection system could be explained as follows (see Figure 3.9):

1. From the data sources we extracted all the necessary data to create the variables.
2. We created a profile of each user, a vector of variable values that defined their behavior up to a certain date.
3. Based on feedback from older campaigns, we ran a number of supervised algorithms in a number of configurations to determine which one could be best for this campaign.
4. We used the chosen model to compute a fraud score (a prediction or probability) for every customer in the target area for a new campaign. We excluded customers that had already been checked in recent campaigns.
5. We created a campaign of a desired size N by selecting the N customers with highest fraud scores.
6. When the campaign results returned from the field, the feedback (verified fraud and non-fraud cases) was added to the system automatically².

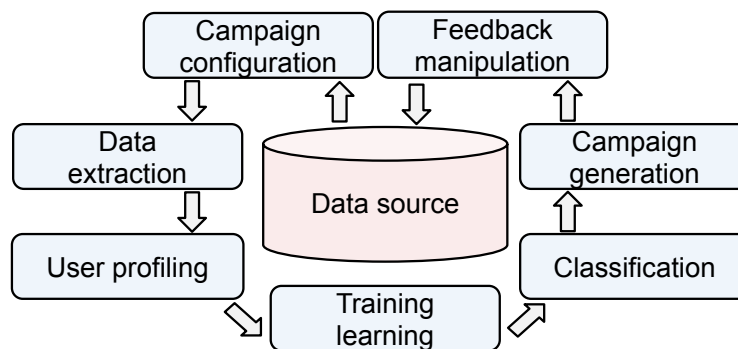


Figure 3.9: The Process of NTL Detection.

This methodology allows iterative learning from the feedback from previous campaigns, as well as the addition of new algorithms and variables. A detailed description of the procedure is given below.

²The company centralizes all the campaigns results introduced manually by each technician in a database, and our system reads this information automatically to update the feedback.

Initial Campaign

The first step in our approach was to create a model able to learn the pattern frauds. This step was needed the first time we created a campaign for a new population for which we had no feedback. It was assumed, however, that some set of verified fraud cases was available from the baseline campaigns by the company, which were labeled P. A set of randomly chosen customers was selected and labeled N, with the understanding that some labels may be wrong. In our case, we needed to do this learning phase on the first campaign for a small city and for the first country-wide campaign.

In all campaigns from the initial one in one population, the training and test sets would contain verified positives and negatives from previous campaigns.

The Campaign Phase

After the customers had been selected for a campaign, each customer had a fraudulent score, and the utility company would select those with higher scores that had not been verified recently, and sent technicians to check the corresponding meters.

The results for each customer could be summarized as *Fraudulent*, *Non-fraudulent* and *Absent*. Fraudulent are those customers who have committed fraud or whose meter does not record (i.e., it does not correctly measure consumption). Non-fraudulent customers are those whose meter could be checked and showed no signs of tampering or malfunction. Finally, absent customers are those for which the technician could not gain access to the meter. Absent customers are a significant fraction of the campaign feedback. We do not include them in our performance calculation or in the feedback to our system (i.e., they are labelled neither P nor N), although it is believed that a fraud among these may be higher than average, because fraudulent customers will try to avoid being inspected. The field reports for the campaign contain a number of distinct codes, some corresponding to malfunctioning meters and some to true fraud; as mentioned, we do not differentiate them in our system and label them all as positive for the feedback.

Finally, the system needed to process the results of the campaign. The profiles of the users from the campaign were stored with their corresponding P/N label. This labelled data would be used as training data in the following campaigns.

3.3.4 Learning from Feedback

After feedback had been incorporated in the system, we had additional verified fraud cases (in addition to those coming from the baseline campaigns) and verified non-fraud cases. These could be used to train new models for further campaigns. Note that we decided *not* to include the baseline fraud cases in the training sets of further campaigns because there was no

guarantee that the profiles *at the time the campaign was performed* were indicative of fraud. The company could have chosen them for inspection based on behavior previous to the records we were given, or based on side information not included in the records. Direct feedback information was considered more reliable.

3.3.5 Algorithmic Details

Many classifier-building algorithms with different configurations were tested. Those that contributed to real campaigns include:

- Naive Bayes.
- K-nearest neighbors: different number of neighbors and distance weight were tested.
- Decision Tree inducers, including C4.5 and CART: both Gini and Entropy split criteria as well as the number of features used were tuned.
- Neural Networks with backpropagation training: The learning rate, momentum, epochs, the number of hidden layers and the number of errors allowed were parameterized.
- Support Vector Machines: both linear and radial basis kernel functions were tested. We also tuned the cost for misclassification as well as the gamma (for the RBF kernel).
- Random Forests: the number of iterations, as well as the parameters tuned in the Decision Tree were modified.
- Gradient descent Decision Tree with CART: besides the optimization applied in the Random Forest, we also analyzed how the loss function (deviance or exponential) modified the performance.
- AdaBoost with C4.5 decision trees, naive Bayes, and PART: the number of iterations were optimized.

The tools used to implement these algorithms were the Knime Analytics Platform [14] and the scikit-learn Python library [101].

3.3.6 Initial Results

Initially, we created fixed-size campaigns, and the scoring of each customer was Boolean (predicted fraud or non-fraud), so we optimized f-measure to balance precision and recall. After the first campaign, we wanted to assign a numerical fraud score to each and every customer in order to create a sorted list of all customers; the company could then choose the size of the campaign going down the list as far as desired. The metric

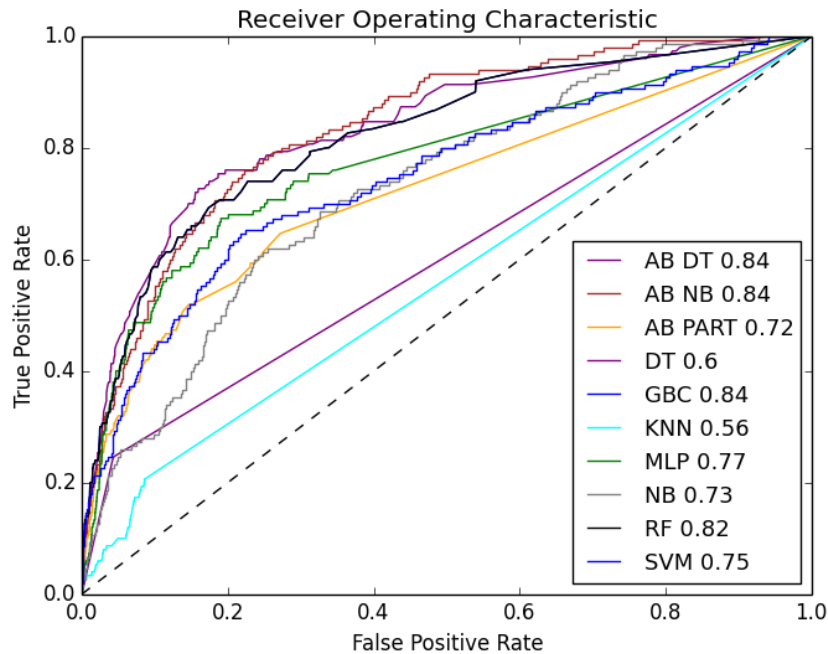


Figure 3.10: Area-under-curve values of the algorithms applied to the 4.5x feedback campaign in 4-fold cross validation. The meta-algorithms (Gradient Boosting Decision Tree, Random Forests, and AdaBoost with Naive Bayes) were the top performers.

to be optimized among configurations was then the Area under Curve, to maximize the position in the list of fraudulent customers rather than P/N hard classifications.

In the initial campaigns, we combined several classifiers, with the final scoring being the voting or average among their individual scores. Furthermore, most algorithms could be run in several configurations (e.g., parameter settings). Our system semi-autonomously³ explored several configurations of each algorithm and several candidate combinations (e.g., including or not each algorithm and assigning voting weights). This model choosing process is not fully automatized, however, it can be automatized using a classification-validation cross-validation process, to facilitate our understanding and post-analysis of the results. In our most recent campaigns, we have opted for a single Gradient Boosting Model, because we observed that it gave better AUC than any ensemble, including other algorithms (see Figure 3.10).

The campaigns were generated once a month, executed in a commercial

³ The choice of configuration was not totally automatized to facilitate our understanding and post-analysis of the results. This process can be fully automatized using the validation-test case with the feedback information easily.

computer (i.e., not a cluster). For this reason, we prioritized the scalability of the software instead of the performance speed. Depending on the input data (i.e., the population or the number of months to compute) the system could last from hardly an hour to a day. With a fair optimization, the software speed could be easily boost.

Our first campaigns were conducted in three medium-sized locations (population between 50,000 and 100,000) to experimentally test the efficiency of our methods to detect electricity fraud without investing on large, costly campaigns. It was soon clear that they were achieving precision levels notably better than the baseline.

To be precise, let us take precision as our main criteria, i.e., fraction of fraudulent users discovered among those inspected. The size of the campaigns was equal to the baseline ones, so we did not increase precision by simply inspecting fewer customers. Then campaigns consisting of randomly chosen customers had a precision of around $z\%$ ⁴, and company baseline campaigns had essentially no better results. Our three initial campaigns had a precision of 12x, 5x, and 5x the baseline.

Encouraged by these results, the company proposed carrying out a test on a nationwide level (Spain) with several million users, also to test the scalability of the approach to detect electricity. We were provided with historical fraud cases from the whole country and returned to the company a list of several thousands of customers sorted by fraud score. The company ran a campaign consisting of the top 10,000, as that was the standard size of their baseline country-wide campaigns. The campaign had a precision of 4.5x that of the baseline. That is slightly less than the worst score achieved in small populations, but of course much more than the baselines. More interestingly, months later the company carried out a second campaign taking the next 10,000 customers from the same list, which had lower scores, so a priori lower precision was expected. Surprisingly, precision was again very close to 4.5x the baseline. As a side-effect, the campaigns provided a dataset of 16,000 customers with reliable fraud/non-fraud labels for further campaigns, the remaining 4,000 being “absent”.

The lower performance of the first country-wide campaign (the one that achieved a 4.5x performance) with respect to the best city-wide ones (12x) merited consideration. We attributed this fact to the higher diversity of customer behavior, energy rate usages, climates, and fraud patterns at a nationwide level. More generally, considering a large user base may blur the patterns that affect only some subsets of customers.

If we break down the data and analyze the results of our campaigns as partial results by the tariff (Figure 3.11), the performances vary notably depending on the tariff; for example, in the 4.5x campaign, the two most common tariffs were those that achieved best performance). This can be read as:

⁴ The exact figure is withheld at the request of the company, but it is a small 1-digit percentage.

- The information of the less common customers is blurred by that of the most common customers.
- We have less information from these customers, the latter being more difficult to profile and detect.

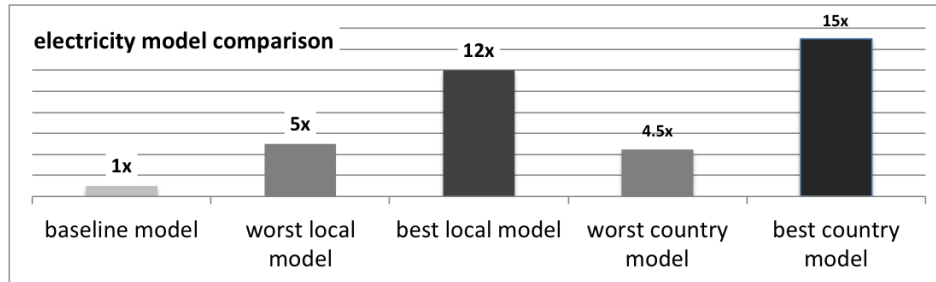


Figure 3.11: Comparison between the precision of the baseline model and our methodology in a local population and in all the country in our electricity campaigns.

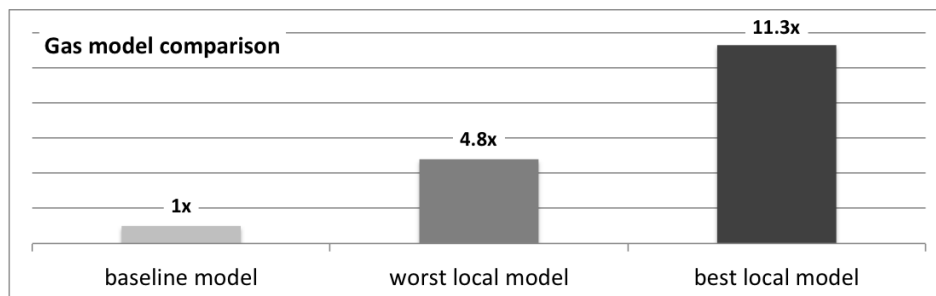


Figure 3.12: Comparison between the precision of the baseline model and our methodology in all the country in our gas campaigns.

3.4 Exploiting the Classification Approach

Our first approach from [36], explained in Section 3.3, provides a vision of our system during the first years of the project. Then, the system evolved to fit the new requirements of the company but also to improve precision. Below we detail the improvements introduced in the system that were explained in [32].

3.4.1 More Labeled Instances

One of the problems in our system was the lack of instances to train models. The use of our own feedback to train models could be insufficient, since many times the campaigns were small. To increase our training dataset,

we started to include all the NTL cases and non-NTL cases from the company, i.e., include feedback from campaigns that were not generated by our system.

To improve our system we also considered it appropriate to make exploration campaigns to feed our system with profiles that are not represented in our campaigns. These campaigns were not carried out because the main objective was not the precise detection of fraud and, therefore, they could be of low success and therefore have low performance for the company. The inclusion of NTL and non-NTL cases not generated by our system could mitigate (though not solve) the classic problem of exploitation vs. exploration in machine learning.

3.4.2 Segmentation of the Campaigns

At first, our system was thought of as a method to detect NTL cases in a specific dataset of similar customers. However, when we implemented the same solution in a more generic campaign (e.g., to detect NTL throughout the country), although it still generated good campaigns, it had an undesired behavior: the system detected NTL only in specific types of customers (e.g., the company has more customers with a contract than customers without a contract, but the proportion of NTL instances in the customers with no contract was higher without a contract). These results suggested to us that it was necessary to segment the customers according to their characteristics:

- Customers with no active contract: Since this type of customer absorbed all the learned patterns from the model (i.e., the characteristics of the dataset and the existing biases made the algorithm only to focus on detecting NTL in customers with no contract), we generated a specific campaign for this type of customers (i.e., we trained a model using only customers with no contract, and scored with this model also the current customers with no contract).
- Customers with an active contract: from the rest of the majority of the customers (around 90% of the customers in the company have an active contract), we sub-segmented the campaign into sub-campaigns depending on the region and the tariff of the customers:
 - Region where they live: Spain is climatically rich. Therefore, there are colder and less sunny regions that would have different consumption behavior than the rest.
 - Tariff: The customer has a tariff aligned to their characteristics and behavior. Analyzing aside each tariff is the best way to single out customers by their consumption patterns (e.g., small apartments, big houses, or industries).

Despite that previous segmentation, we needed some extra segmentation due to the existing dataset-shift (Formally, if $P_{population}(x)$ and

$P_{labeled}(x)$ denote the real population and labeled (train) fraud distributions, it often happens that $P_{labeled}(x) \neq P_{population}(x)$, since $P_{labeled} = P_{population}(x|s = 1)$, where s is the binary condition of visit (i.e., if the customer was visited) in some segments: the company's labeled instances (i.e., the results of the campaigns from the last two years) do not reflect faithfully the distribution from the company's customers, since this strongly depends on the historical success of the campaigns conducted before in that particular segment. This imbalanced training dataset generates biased models (see for instance Figure 3.13). For this specific case of imbalance, we created specific campaigns, insulating these customers that are over/under-represented. That is:

- Over-representation campaigns: The over-representation of a type of customer can absorb all the classification capabilities (i.e., that the system only learns to classify that type of customers). A consequence of this is that only the over-represented type of customer receives a high score. To avoid this, isolating these customers and generating a tailored campaign for them avoids the imbalance.
- Under-representation campaigns: There are cases where the information from a type of customer is not good enough to detect fraud (e.g., there are no positively labeled cases) and therefore that type remains unconsidered. Generating specific campaigns for this type of customer, even though the labeled information is not extensive, forces the system to detect NTL patterns for this under-representation.

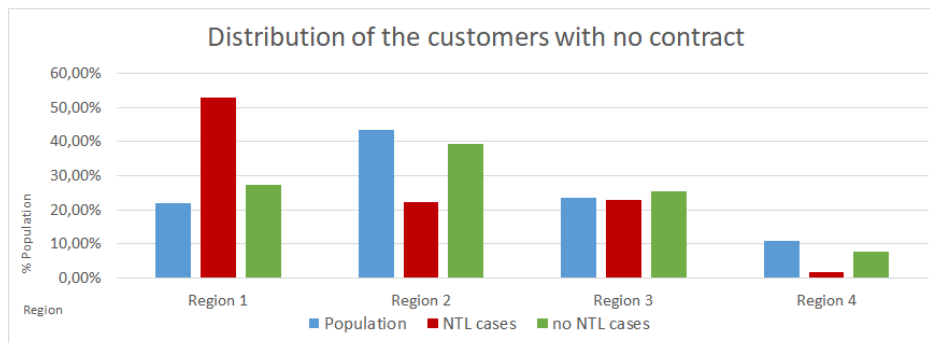


Figure 3.13: Real example of the dataset-shift problem between the proportions of the real domain (customers from Spain with no contract in November 2018) and the labelled instances of the customers with no contract (feedback available from the same type of customer since October 2016); in the region of Madrid the company has more cases of NTL than in any other region, and this led our system to over-estimate the score assigned to the customers in that region. This was solved through a segmentation, generating two different campaigns (Madrid and the rest of Spain).

Contract Status		Region		Tariff
Customers with contract	⊗	Madrid	⊗	0-10kWh (w/ or w/o hour discrimination)
		Castilla la Mancha		10-15 kWh (w/ or w/o hour discrimination)
		Galicia		>15kWh (big customers)
Customers with no contract		Castilla y León		>>15kWh (industries)
		Customized		Customized

Figure 3.14: Our system can be fully configured. The contract status, the region and the tariff of the customer can be set before starting the process, allowing dozens of different configurations, generating successful tailored campaigns.

3.4.3 New Features

The evolution of our NTL detection system made us evolve and adapted the features to flexibilize them and facilitate the optimal profile of different types of customers. These are the main changes in comparison to the features explained in Table 3.1:

- **Contract Status:** To represent customer behavior in accordance with the contract status (active contract or not), we included more consumption-related features for the customers with contract. For the customers without an active contract, as they do not have a recent consumption record, we only included a reduced number of consumption features that focused on their consumption behavior just before they cancelled the contract (with the aim of detecting abnormal behaviors).
- **Flexible Features:** Instead of delimiting the features to a closed meaning e.g., considering that the abrupt decrease of consumption (the abrupt decrease in consumption are a family of features that aims to detect consumption drops in consecutive months.) feature consists of a binary value indicating a reduction of X%, we opted to include as much raw information as possible, e.g., in the previous example, including the raw value of the ratio between the consumption of the last two consecutive months (i.e., representing this behavior with a continuous value). This would give the algorithm more flexibility than using static binary features, guaranteeing that:
 - The continuous value provides a richer representation of the ratio of consumptions in comparison to a binary feature and, therefore, the partition process from the Decision Tree would be better.
 - A richer representation of the reality also provides the system with other information that simple binary features cannot. For instance, it could be true that in certain segmentations, a high

increase of consumption in two consecutive periods of time would be an abnormal behavior too. The continuous feature can provide this information to the system.

- The flexibility would help the system to be successful over time, even though the consumption patterns from the customer change. This can be exemplified with what we regard as low consumption; in the future, apartments will be more energy efficient, thus what we consider now a low consumption in the future could be the average consumption and, for this reason, including raw information of the consumption instead of binary information (e.g., the customer has consumed less than 2000 KWh in the last 12 months) will make the system more future-proof.

In Figure 3.15 there is a visual example that explains the flexibility desired in our system.

- Different versions: to boost the information given to the algorithm, different versions of the same concept are provided to the algorithm. This will allow more flexibility for deciding which feature can give the maximum information during the training (e.g., there are several abrupt decreases in consumption features, with different nuances; depending on when that consumption drop happened, how many months are considered in the consumption decrease, etc.).
- We also reduced the number of features used in the system to avoid overfitting.
- Categorical Features: we exploited the categorical information available from the company's database.

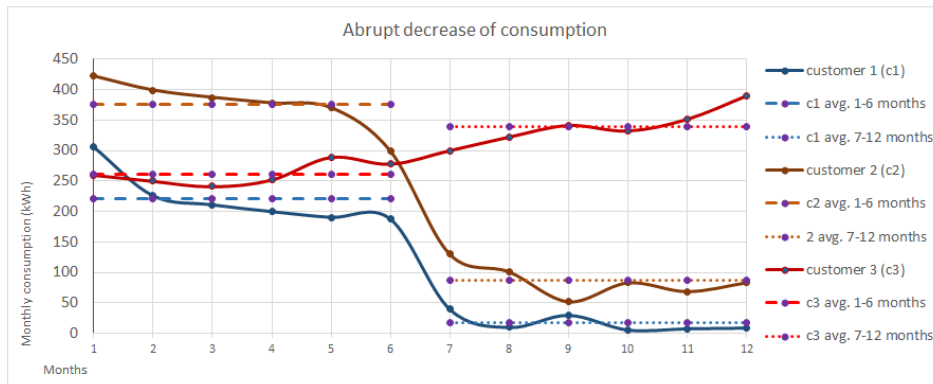


Figure 3.15: Providing to the system the ratio between the consumption of two consecutive periods of time (i.e., 0.07 for customer 1, 0.023 for customer 2 and 1.3 for customer 3) instead of a binary feature such as "during the second period the customer consumes lower than 10% of the consumption in the previous period" gives more information and flexibility to the system. The system can determine the proper split in the training stage, and moreover we provide extra information (e.g., that customer 3 is consuming more in the second period, something that would require an extra feature if we used the binary feature).

Despite these changes, the basis of the variables is the same: to generate a set of consumption variables that can represent the anomalous consumption typical of fraud, and a set of complementary variables (the static, visiting and sociological variables) that complement and contextualize the consumption variables.

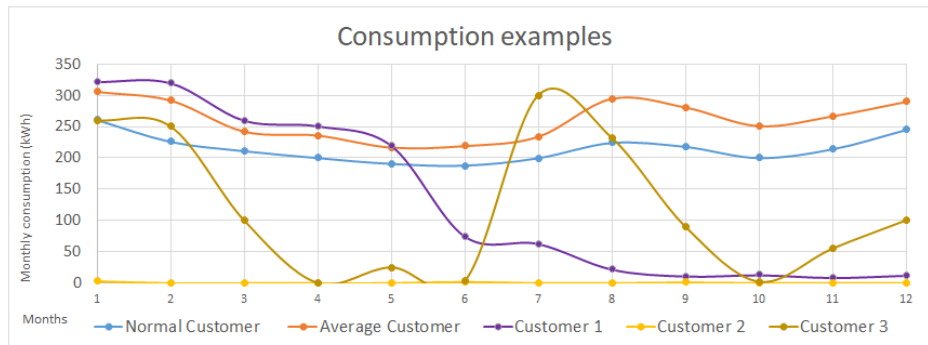


Figure 3.16: Some examples of different consumption behaviors. The features need to provide enough information to extract as much information as possible from these profiles: Long periods of low consumption (customer 2), abrupt decrease of consumption (customer 1), similar consumption behavior to the expected consumption curve (Normal customer) or an abnormal consumption curve (Customer 3).

3.4.4 Algorithm and Metrics

To simplify our system, we went from combining different algorithms at the same time to using only a Gradient Boosting Decision Tree (GBDT) model. This decision was multifactorial, being the optimal decision if we analyzed the situation in terms of accuracy, simplicity, flexibility, transparency, execution time and available resources.

In terms of accuracy, GBDT algorithms are the state-of-the-art solutions when working with tabular data. Moreover, these good results can be achieved efficiently: modern GBDT implementations require less computational resources than other solutions, scale well, and do not require deep preprocessing to obtain optimal results (e.g., XGBoost allows empty values, as well as LightGBM and Catboost allow categorical values). All of this simplified the automatic generation of campaigns, making it easier for us to generate campaigns quickly in various domains. Moreover, as we analyze in further chapters, there exist fast and accurate methods to achieve robust explanations in tree models.

The alternatives considered were the Support Vector Machine and Neural Networks. The former was rapidly discarded because of computation time ratios (it is much slower than GBDT, especially if a non-linear kernel is used in the SVM). The latter was also considered (especially due to its popularity) but discarded for many reasons. First, deep learning is also slower than GBDT (and the lack of GPU capabilities in the company's servers made this slowness insurmountable). This is especially true considering that deep learning does not provide high performance out-of-the-box, requiring deep parameterization (i.e., unique configuration of layers, connection and activation of neurons per campaign generated). Furthermore, the theoretical superiority of deep learning in terms of accuracy in text and image classification problems is not seen in tabular data, where GBDT models achieve (at least) on par accuracy, with much less effort. This is analyzed in this recent paper [126].

Regarding the metrics used to evaluate our system, we started using the f-score, but then we considered the use of the ROC-AUC score to avoid the use of thresholds, since the company usually considered the order of the customer instead of the score itself to build the campaign (i.e., the company included the top-k customers in a campaign instead of including the customers with scores higher than a threshold). Finally, we used the PR-AUC instead of the ROC-AUC score since it provides more robust results with an imbalanced dataset, as explained in Section 2.1.2.

3.4.5 More Results

The modifications explained in this section helped us to generalize the use of the system, making it flexible to detect NTL cases for different customers. We exemplify this success with two specific campaigns, the 36% of accuracy in customers with no consumption and the campaigns

generated to detect NTL cases in customers with no contract.

Customers with no Consumption After several years of generating campaigns, the company shared with us their concerns about the customers with no consumption. After the installation of smart meters, they considered that many of the customers with no consumption (i.e., with no difference according to the difference between meter readings) would decrease. However, it was increasing month after month.

To solve this problem, they started to create rule-based campaigns to control this type of customer. For instance, to visit those customers that according to the company had both electricity and gas, but the consumption of gas did not match the electricity consumption. In general, we considered the approach taken by the company to be excellent, visiting those customers with very abnormal behavior, but the results were bad. In part, the poor results could be explained by the centralizing power that Madrid has for the inhabitants of the Castilian territories: many people are compelled to live in Madrid and therefore there are many cities and towns in Castilla la Mancha and Castilla y Leon with descending populations and, therefore, the number of empty houses and apartments is increasing in these regions.

For this reason, they asked us to fit our system to detect NTL cases in this specific type of customers. The campaign generated was trained by using as labelled instances those customers that would be defined as customers with long periods of no consumption, i.e., customers that had 0 or close to 0 (up to 10kWh) in at least the last 7 months. The campaign was conducted in 4 regions, where the best campaign achieved up to 36% of precision, and the least accurate achieved a precision close to 10%.

Customers with no Contract The company used to generate fairly successful rule-based campaigns for the customers with no contract, especially those that were recidivist, i.e., power supply points with constant fraud (e.g., houses occupied by squatters). In spite of the good results, we adapted the NTL detection system to generate campaigns for the customers with no contract.

Our system has achieved several times campaigns with a precision higher than 50%. Moreover, we generated campaigns to detect NTL cases for large customers (e.g., industries), recovering a large amount of energy.

3.5 Overall Analysis of the Classification Supervised Approach

The supervised approach to solve the NTL detection problem was successful in general, but had different problems that needed to be tackled. Below we set out these problems, all of them related to each other.

Lack of Robustness Our system achieved high prediction in NTL cases, but lacked in robustness. For instance, in many cases the results in distinct but very similar campaigns were very different (e.g. precision of 20% and 2%). In part, this could be exemplified in table 3.3, where we analyze the results of the system in a test dataset using 4 different metrics: the precision, the F1-Score, the AUC-ROC score and the Average Precision Score in one dataset. As we can see, if we focus on the results in precision, the results are excellent. However, benchmarking only provide a biased vision of our system: The results in F1-Score, and Average Precision determine that our system might not generalize as desired.

DOMAIN	ACCURACY	F1-SCORE	PRECISION
Domain D_{AN}	0.98	0.58	0.64

Table 3.3: Results from a test in a labelled dataset, with an 80% training - 10% validation - 10% test distribution. The results in the table, which correspond to the results in the test, indicate a priori a good benchmarking (especially in terms of precision), but the low performance in terms of F1-Score and Average Precision indicate that our system is not robust enough.

Our efforts to increase the robustness by increasing the regularization to reduce the overfitting or the introduction of new labelled instances to improve the representativity of the labelled instances did not improve the overall robustness.

Amount of Energy Recovered In table 4.1 and Figure 3.17 we show another side of our system’s lack of robustness: the system learn patterns to detect NTL cases with low energy recovered. In general, patterns that might be good to achieve good results in training/validation tests do not properly generalize in real scenarios.

DOMAIN D_{AN}	$energy_{528}$	$energy_{211}$	$energy_{106}$	$energy_{42}$
Reference	1112625	798198.3	582480.8	366088.1
Classification	434531	196407	97659	37838

DOMAIN D_{AN}	$NDCG$	$NDCG_{528}$	$NDCG_{211}$	$NDCG_{106}$	$NDCG_{42}$
Classification	0.52	0.25	0.16	0.11	0.07

DOMAIN D_{AN}	$median_{528}$	$median_{211}$	$median_{106}$	$median_{42}$
Reference	1324	2704	4131.5	6884
Classification	692	782.5	735	610

Table 3.4: A comparison between a perfect model and the classification model in terms of $energy_n$ (i.e., amount of kWh recovered in each threshold n). As we can see in the results, we recover a low amount of energy, an indicator that we might not be learning the optimal patterns that represent the NTL.

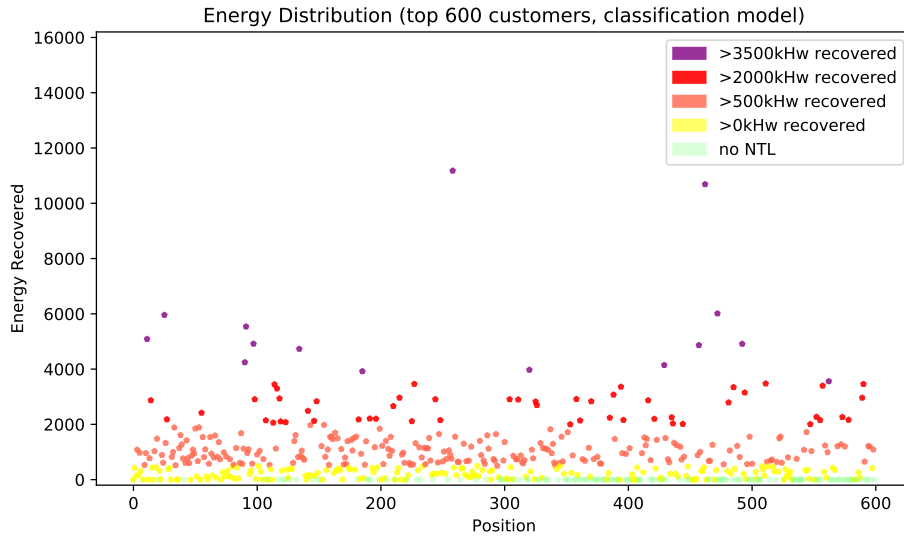


Figure 3.17: Results from a test in a labelled dataset, with an 80% training - 10% validation - 10% test distribution. The model correctly predicts the NTL cases, but as exemplified in table 4.1, we are detecting NTL cases with a low amount of energy to recover.

Interpretability and Bias Detection Our approach to validate the features built in our system was, in the first place, to analyze the odd-ratio of the features (see Table 3.5 from [30]).

VARIABLES	ORPG	ORPN
Abrupt decrease of consumption	18.6	3.4
Long period of low consumption	6.2	3.0
High consumption discrepancy	10.4	2.4
High range higher lower consumption	12.2	2.0
Gas consumption without electricity	11	2.9

Table 3.5: Significant odds-ratio of some features from the electricity campaigns. Both the odds-ratio between the fraudulent and non-fraudulent customers (ORPN) and the odds-ratio between the fraudulent customers and the customers not part of any campaign (ORPG) are included.

However, this approach is poor, since it only analyzes the values of the features (helping to detect, for instance, biases in the labelled instances), but does not analyze their influence on the system. Therefore, it is difficult to understand the lack of robustness and energy recovered problems when the patterns learnt by the model are not easily interpretable.

3.6 Discussion

This chapter explains our first approach to the NTL detection problem using data science techniques, being the basis for the future chapters of

this part of the thesis. The proposal is very similar to others existing approaches in the literature (i.e., a supervised classification method that predicts using a black-box algorithm), achieving good results.

However, in contrast with other works, we offer a critical analysis of our work. In Section 3.5 we explain the existing problems of our system: the lack of robustness, the low amount of energy recovered per NTL detected, and the lack of interpretability. These challenges are analysed and mitigated in the following chapter.

Chapter 4

A Regression Approach to NTL Detection

4.1 System Goals and Challenges

The system explained in previous Chapter 3 has been successful as an NTL detection system. Nevertheless, several problems were detected, as we discussed at the end of the Chapter (Section 3.5). Here we provide a more extensive and ordered explanation of the existing challenges.

Technical Challenges

In general, our system has achieved good results, especially considering that it is implemented in a European region with a very low ratio of NTL cases. However, the robustness of our system campaigns varied depending on the type of campaign. For instance, our system is accurate in certain types of campaigns where the type of customer was predefined (e.g. customers with no current contract¹, or customers with long periods of no consumption²). However, in more generic campaigns (i.e. campaigns that included hundreds of thousands of customers) the system underperforms in robustness, i.e. the system cannot consistently provide good results.

According to our experience and knowledge, two fronts explain these problems: the existing biases in the labelled instances available from the company and the difficulty of properly benchmarking a model using a validation dataset.

¹ Customers with no contract refers to the customers that had a contract in the past, but the contract is currently cancelled. In many cases, these customers maintain the wire and meter installation and, therefore, can commit fraud. Our system has achieved many campaigns of around 50% of precision.

² As people in Spain move to cities, many villages become empty. This is a problem for the company as they do not know how to differentiate a house without consumption because it is a second home with punctual consumption or a fraudulent client. Our system was able to detect NTL cases for these types of customers with a precision of up to 36%.

Regarding the data-related problems, we have already explained in [32] how we detected different types of biases and other data-related problems in our data. These problems are a direct consequence of using observational data produced for other purposes. Therefore, the available information does not reliably represent reality, and it is a challenge to ensure generalizability since the assumption that the labelled and the unseen instances are i.i.d, i.e. independent and identically distributed, is not met. For instance, the fact that the company visits more customers suspected of NTL leads to an over-representation of these customers, meaning that *average* customers with a normal consumption are grossly under-represented in the system. A similar problem is that the company generates more campaigns in those regions where it has historically achieved better results, making the quality of the labelled information in under-visited regions very low. Therefore, it is a challenge to continually build robust models when the labelled dataset does not correctly represent reality.

Our first efforts consisted of implementing classical machine learning techniques, e.g. to modify the model's regularization and tuning, but no improvement in the campaigns was observed. Similarly, we attempted to improve the labelled information used to train the model, e.g. by weighting the customers according to their representativeness, balancing the class imbalance typical of fraud detection problems, or implementing a cost-sensitive solution. However, after applying these solutions, the results were inconclusive: some of the experiments validated in our labelled information had initially unsuccessful results in real campaigns. Moreover, the company's demand for having short-term results made us rule out the generation of exploratory campaigns with these techniques that could offer us a long-term improvement of the system. All of this evidenced the difficulty of benchmarking our NTL system on validation datasets and a scalar metric [48].

At this point, we discarded the most complex methods and introduced some simple solutions that could be easily validated. For example, in [32] we explained how we segmented the customers to build more targeted campaigns to mitigate imbalance-related problems. For benchmarking, we used the Average Precision Score³, which provides a good generic vision of how well a model ranks, without setting a threshold when the data is highly imbalanced [42]. These solutions improved our system. Nevertheless, the system was still not sufficiently reliable for its industrial adoption.

Economic Efficiency

The use of machine learning solutions to generate campaigns is justified if it provides a better solution than a random selection of customers or a

³ The Average Precision Score is the scalar value that results from summarizing a precision-recall curve as the weighted mean of precisions at each threshold, using as weight the increase in recall from the previous threshold.

baseline non-smart method (e.g. a basic rule system consisting of visiting those customers that have had an abrupt decrease of consumption). The term *better solution* includes different aspects from the company's point of view but can be summarized in the following two dimensions:

- The machine learning solution is more precise than other solutions, i.e. the proportion of True Positives is higher than the random selection or the rule-based approaches.
- The machine learning solution recovers more energy than other solutions, i.e. the energy estimated that the NTL cases have not paid (and should be charged in the near future to those customers) is higher than the energy recovered from random selection or rule-based campaigns.

Therefore, a campaign with a low precision but a large amount of energy recovered would be considered a successful campaign. Similarly, a campaign with fairly low energy recovered would also be considered a good campaign if many NTL cases are discovered, as it would prevent energy loss in the future. Understandably, an excellent campaign would be able to combine both good precision and a high amount of energy recovered.

To better understand what would be considered a good campaign in terms of energy recovered, it is necessary to note that the average annual electricity consumption per household in Spain is about 3500kWh. In addition, the distribution company can legally invoice the NTL for one year: "*... the distribution company will invoice an amount corresponding to the product of the contracted power, or that should have been contracted, for six hours of daily use during one year,...*"⁴. Under these circumstances, the following classification has been considered for the purpose of analysing the NTL cases detected according to the energy recovered:

- >3500kWh recovered: The detection of these customers is a priority due to the amount of energy lost.
- Between 3500kWh and 2000kWh recovered: These NTL cases are also important. As in the previous example, the consumption curve should reflect an abnormal behavior that the predictive system should be able to detect, e.g. a long period of low consumption.
- Between 2000kWh and 500kWh recovered: These NTL cases should have some abnormal consumption behavior (e.g. a recent abrupt decrease of consumption). However, their detection should not be prioritized over the customers with an NTL case estimated to recover energy >2000kWh.

⁴ Real Decreto 1955/2000, de 1 de Diciembre, art. 87

- 500kWh or less: Although these are NTL cases, their consumption behavior might not properly represent the NTL behaviour (e.g. an abnormal consumption curve or an abrupt decrease of consumption). Therefore, these NTL cases might not be prioritized over the previous NTL cases, as they might include in some cases noise or biases in the system.

From the company’s point of view, our system tended to detect NTL cases with low energy to recover. For this reason, some machine learning techniques were implemented to increase the amount of energy to recover (e.g. weighting the customers according to the energy recovered). However, the results obtained after applying these solutions were inconclusive and, in many cases, seemed to aggravate some of the existing data biases (e.g. by oversampling the customers from specific regions).

System Transparency

Although it is generally accepted in the literature that the black-box algorithms are more accurate than other more interpretable approaches, their use poses a clear problem in terms of transparency, which greatly hampered the development of our system. The problems explained above and the lack of conclusive results in our tests were a direct consequence of the impossibility of understanding how the methods implemented impacted our system.

This lack of transparency affected the company’s stakeholders in different ways. On the one hand, the stakeholders historically in charge of generating the NTL campaigns could not validate the patterns learned by the model. As widely analyzed in the literature [100, 99, 11], the supervised methods only detect correlations, and therefore human supervision is necessary to validate them as reliable causal patterns (or, at least, reliable correlations in the company’s context). The use of a black-box algorithm made this task challenging, so they could neither easily detect undesired patterns nor suggest system improvements. On the other hand, managers in charge of setting company guidelines had to make decisions regarding the use of the system (i.e. whether to have confidence in the system and use it to generate campaigns) in a blind manner, based solely on their results.

As explained in more detail in Section 4.3, our first approaches (i.e. to use Feature Importance and LIME) to provide explainability to our system (and therefore to make our system more transparent for the stakeholders) were insufficient.

4.2 The Regression Approach for NTL Detection

4.2.1 From Classification to Regression in NTL Detection

The classification and regression models are two supervised methods that can be defined as follows: being X the labelled instances $\{(x_1, y_1), \dots, (x_n, y_n)\}$,

where x_i is the feature vector that represents an instance and y_i the value to be predicted, the supervised model aims to learn the function $f, Y = f(X)$, wherein a classification model Y is either 0 or 1 (or $0 \leq Y \leq 1$ if the model provides probabilities), and in a regression model the value to predict is continuous (i.e., $Y \in \mathbb{R}$).

The classification approach to detect NTL is widely seen in the literature (see, for instance, the examples from Related Work, Section 3.2 or our work explained in [36]). This approach, despite the good results that it can achieve (in [32] we explain how we have achieved campaigns with an accuracy higher than 50%), oversimplifies the representation of the reality in our NTL detection system since it equalizes the importance of each NTL case: both the customer that has been committing NTL for one year and has stolen 3000 kWh and the customer that had a meter problem for a few weeks (and therefore the energy loss is low), have the same label, even though the former case is much more important for training a supervised model for NTL detection. The higher the energy recovered, the better, as already introduced in Section 4.1, is true for several reasons.

- On equal terms, it is preferable to recover more energy at once in each visit from an economic point of view.
- The company usually detects short-term NTL cases through smart meter sensors. That is, if the smart meter detects a manipulation, it sends a signal to the company to warn about that manipulation, taking some days (or weeks) to include that customer in a campaign. Focusing on detecting these cases through data analysis may overlap the sensor NTL detection method. However, the long-term NTL cases are NTL cases that remain undetected.
- The company might have problems recovering all the NTL from long-term fraudulent customers due to legal reasons. For this reason, companies focus their efforts on detecting these long-term fraudulent customers to reduce the difference between the energy loss and the energy they will be able to bill⁵.

Moreover, as we explain in Section 4.1, we work with observational data, i.e. data produced for other purposes that has not been prepared nor randomly sampled to properly represent the actual customers. The fact that the labelled information available corresponds to customers visited to control abnormal behavior (or correct a meter problem), altogether with other company-related decisions that aim to maximize the campaign results (e.g. the companies usually over-control the customers that constantly commit fraud), lead the training dataset available to train the model to not

⁵ For instance, not detecting a long-term NTL customer (e.g. 20 months of energy loss) will increase the energy stolen by the customer. A customer that has been committing NTL for three months will also steal energy, but the company will still be able to bill all the stolen energy if it is detected during the next nine months.

represent the reality of the company’s customers properly, disserving the machine learning process. Consequently, we are dealing with the existence of dataset-shift, i.e. the joint distribution of inputs and outputs differs between the training and test datasets: if $P_{population}(x)$ and $P_{labeled}(x)$ denote the real population and labeled (train) fraud distributions, it often happens that $P_{labeled}(x) \neq P_{population}(x)$, since $P_{labeled} = P_{population}(x|s = 1)$, where s is the binary condition that indicates if the customer is included in the training dataset, in our case if the customer was visited. All these problems cause the robustness degradation of our classification approach, visually represented in Figure 4.1.

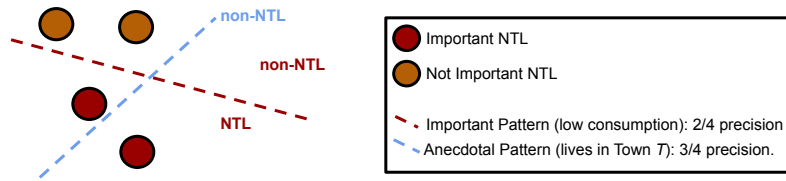


Figure 4.1: With the binary classification we are equating the importance of each NTL, learning undesired patterns: if we do not prioritize the darker red instances (NTL cases with a large amount of energy recovered and, therefore, better representatives of the behavior of an NTL case), we might prioritize undesired patterns like the one represented in a blue pattern. The result is a biased model that cannot robustly detect NTL cases.

In this work, we propose to use the energy to recover as the value to be predicted by the model, i.e., to convert our classification approach with a LogLoss function model (explained in Section 2.1.2) into a regression problem, where the value to predict is the amount of energy recovered in the NTL case. With this fundamental change, we aim at improving our system by focusing on learning better patterns that generalise better on unseen data, as we explain below:

- By breaking the NTL/non-NTL binary representation of the NTL case, we implicitly indicate to the system that it should focus on learning patterns from high NTL cases whose profile should have clearer abnormal consumption feature values (e.g. low consumption during the last year).
- Moreover, we avoid learning patterns from over-represented customers in the observational data due to business-related decisions (e.g. the recidivist customers) if it does not entail greater energy recovery.

If we look again at the example in Figure 4.1, using the energy to recover as the target variable means that the system is going to learn the important pattern first rather than the other.

The two most typical regression Loss Functions are the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), both explained in Section 2.1.2. The difference between the RMSE and the MAE



Figure 4.2: Twenty-four months consumption curve from a recidivist customer that has committed fraud three times (each vertical line corresponds to the moment the company detected that the customer was committing fraud, with the amount of energy recovered). A binary approach would label each case equally (i.e. as a positive instance), overlooking the fact that each NTL detection is different, and needs to be contextualized. The RMSE regression approach would set the desired priority.

loss function is the square of the errors, i.e. the higher errors have more weight in the RMSE (as exemplified in Figure 4.2). Therefore, the RMSE fits better in ranking problems, in recommender systems, or in our purpose of learning patterns from the higher NTL instances from our training dataset.

4.2.2 Experiments: Classification vs. Regression Benchmarking in Real Data

In this subsection we compare both the classification and the regression model for NTL detection and confirm the expected benefit of using regression when the organization’s aim is to recover energy without visiting too many customers.

Preliminaries

Data For the experiments, we will use four different datasets from two regions (*A* and *B*), with two different tariffs (1, the most common tariff

for houses and apartments in Spain, and tariff 2, an equivalent tariff to 1 but with hour price discrimination. The regions are anonymous to protect the privacy of the data.⁶). The customers must have less than 10kwh of Contracted Power to be on these tariffs. The domain D_{A1} (i.e. the customers from region A and tariff 1) has more than 1000000 customers, and domain D_{B2} has less than 50000 customers. The other two datasets fall between these two datasets in terms of population. The proportion of the NTL cases in each domain is lower than 5%. We have around 300000 labelled instances for the D_{A1} domain, several thousand cases for D_{A2} and D_{B1} , and several hundred cases for D_{B2} .

Model For the classification and the regression predictions, we have trained two different CatBoost models. Each model is trained using the same 80% of the positive instances and 80% of the negative instances. We split in half 20% of instances left, keeping the positive/negative ratio, to build the validation dataset (i.e. the data used to tune the model), and the test dataset (i.e., the training, the validation and the test dataset are stratified). The random partition is chosen over considering the timestamp (e.g. the last 10% of NTL cases as the test dataset) to guarantee diversity and reduce the differences between the datasets due to company decisions. To avoid overfitting, the metric used for early stopping to establish the optimal number of trees is the Average Precision Score for the classification model and the RMSE for the regression model. Both models use the same customer profile, with the only difference that for the classification approach we use a binary target (NTL/non-NTL), while in the regression approach we use the amount of energy to recover (information that is provided by the technician when an NTL is detected).

Benchmarking A good benchmarking metric to use if we aim at recovering more energy in our campaigns is the Normalized Discounted Cumulative Gain ($NCDG_n$) [70]. It is a measure of ranking quality that evaluates our output’s correctness with a value between 0 and 1 (1 being the perfect order of the NTL cases, and 0 otherwise). This metric allows us a global vision of the correctness of the predictions made, without considering one specific threshold (i.e. the top 100 customers): in many cases, the number of customers to be included in a campaign is unknown when the campaign is being built.

The $NCDG_n$ is defined as

$$NDCG_n = \frac{DCG_n}{IDCG_n}$$

⁶ A tariff with price discrimination involves charging a different price for the electricity depending on when the electricity is consumed. More specifically, electricity would be cheaper at night but more expensive during the day. The potential customer of this tariff is the customer that has an electric car and charges it at night.

where DCG_n is defined as

$$DCG_n = \frac{\sum_{i=1}^n Rel_i - 1}{\log_2(i + 1)}$$

Rel_i being the relevance (i.e. the score in the ranking, in our case the amount of energy recovered), and $IDCG_n$, i.e. the *ideal DCG*, corresponds to a perfect ordered DCG for the top n elements of the list.

In addition to the $NDCG_n$ metric, we use the amount of energy recovered from the top n scored customers to compare approaches. In both cases we provide four different results (i.e. four different n threshold values): $n = (\text{NTL cases in test})/2$, $n = (\text{NTL cases in test})/5$, $n = (\text{NTL cases in test})/10$ and $n = (\text{NTL cases in test})/25$; each threshold aims to represent different types of campaigns: from very small campaigns where just a few customers are visited to big campaigns where hundreds of customers are included in the campaign.

Benchmarking Results

In Table 4.1 we report the comparison, in terms of energy recovered and $NDCG$ metrics, for the regression and classification approach in the four datasets (for each n threshold).

In terms of $NDCG$, the regression models always score better than the classification models, meaning that the regression approach is able to order better the test customers according to its consumption. Therefore, we recover more energy at the very top of the list, confirming in terms of benchmarking its superiority over the classification approach. This superiority is especially true for small campaigns, where the $NDCG$ value for the classification approach is extremely low.

In terms of energy recovered, the regression approach is superior to the classification approach; the amount of energy recovered in our results is usually higher than the energy recovered with the classification models, especially for small-sized campaigns. Recovering more energy is the desired outcome: accumulating very high NTL cases at the very top of the list would allow the company to generate more fruitful campaigns.

With large or medium-sized campaigns, the benefits in terms of $NDCG$ and energy recovered of the regression approach is not as clear as in small-sized campaigns, as we can see in Figure 4.3: the regression model ranks higher the high-NTL cases (i.e. the NTL cases in which more energy can be recovered, in purple and in red) in comparison to the classification model, but then this advantage fades slightly, and the energy recovered by both approaches becomes more similar.

Energy recovered from an n-sized campaign (kWh)				
Domain D_{AN}	$n = 528$	$n = 211$	$n = 106$	$n = 42$
Reference	1112625	798198.3	582480.8	366088.1
Classification	434531	196407	97659	37838
Regression	468496 (+7%)	267121 (+36%)	164814 (+69%)	73092 (+93%)
Domain D_{AD}	$n = 186$	$n = 74$	$n = 37$	$n = 15$
Reference	362877.4	273622.2	204201.6	139045.6
Classification	164509	68391	39941	8704
Regression	151844 (-8%)	96520 (+41%)	70022 (+75%)	54988 (+532%)
Domain D_{BN}	$n = 79$	$n = 31$	$n = 16$	$n = 6$
Reference	146245.7	102029.7	75141.3	46690.3
Classification	50596	22164	10079	3542
Regression	67163.9 (+33%)	25764.2 (+16%)	15148.2 (+50%)	12595.2 (+256%)
Domain D_{BD}	$n = 19$	$n = 7$	$n = 4$	$n = 1$
Reference	46482.3	31957.3	22607.3	7555
Classification	16799	7472	5975	2691
Regression	14036 (-16%)	11370 (+52%)	8679 (+45%)	5484 (+104%)

Ranking quality from an n-sized campaign (NDCG)					
Domain D_{AN}	$NDCG$	$NDCG_{528}$	$NDCG_{211}$	$NDCG_{106}$	$NDCG_{42}$
Classification	0.52	0.25	0.16	0.11	0.07
Regression	0.57	0.32	0.26	0.23	0.18
Domain D_{AD}	$NDCG$	$NDCG_{186}$	$NDCG_{74}$	$NDCG_{37}$	$NDCG_{15}$
Classification	0.43	0.25	0.15	0.11	0.05
Regression	0.65	0.46	0.44	0.45	0.49
Domain D_{BN}	$NDCG$	$NDCG_{79}$	$NDCG_{31}$	$NDCG_{16}$	$NDCG_6$
Classification	0.45	0.22	0.15	0.11	0.08
Regression	0.47	0.29	0.19	0.16	0.17
Domain D_{BD}	$NDCG$	$NDCG_{19}$	$NDCG_7$	$NDCG_4$	$NDCG_1$
Classification	0.47	0.30	0.24	0.25	0.36
Regression	0.59	0.39	0.43	0.46	0.73

Table 4.1: The table at the top compares classification and regression in terms of energy recovered (i.e. the kWh recovered in each threshold n). As we can see in the results, the regression approach can recover more energy than classification in most cases. In several cases, the amount of energy recovered is significantly greater, especially when the n threshold is small. This means more efficient campaigns in economic terms. The table at the bottom provides a similar analysis, comparing the campaigns in terms of $NDCG_n$. In this analysis, the regression results always outperform classification results in ranking performance (i.e. sorting the customers according to their NTL).

4.3 Analysing NTL Detection Beyond Benchmarking



Figure 4.3: The results obtained in Table 4.1 are confirmed in these images: the regression model recovers more energy at the very top of the test prediction list. More specifically, we can see how the purple cases (NTL cases with more than 3500kWh, the average customer’s energy consumption per year) in the regression model are recovered at the very top of the rank.

4.3.1 Classification vs. Regression in Terms of Explainability

The results from Section 4.2.2 suggest that the regression models recover more energy than classification. However, as explained in Section 4.1, we are not confident with only benchmarking our models: increasing the accuracy in validation sets that are subsamples of biased labelled instances does not guarantee that the system is fair (i.e. the system is unbiased against a particular type of customers, e.g. customers from poorer regions),

and robust (the system will perform as expected in reality, learning causal patterns, with no data leakage [72] nor dataset-shift [105]). The regression approach should be humanly validated as a better method (e.g. learn better patterns) than the classification approach. The purpose of this section is to illustrate this through explanatory algorithms.

The first explanatory algorithm tested in our system was the Feature Importance method. This approach was useful for us to detect biases (e.g. by detecting features that were not indicators of NTL but were too important in the model), but only provided a global vision of the model, with no possibility of analyzing the importance of the features on specific customers with a high score. For this reason we explored the use of LIME to explain our predictions at instance level. As we explain in [31], we were able to implement a rule-based double-checking method in campaigns to discard customers for whom, despite a high score, the explanation obtained from LIME was undesired (e.g. the patterns explained by the local model would not be validated by a human expert). Despite the good results we did not implement LIME as our explanatory algorithm due to the well-known problems of robustness (e.g. [7]) because of the random component of the algorithm but also the difficulty of having an optimal configuration.

After these two initial unsatisfactory approaches, we started to use SHAP (more specifically, the tree SHAP implementation [81] to obtain the Shapley values from Tree Models). According to our experience, the tree SHAP was the optimal approach to obtain an explanation from a Tree Model because of the following advantages summarized below:

- Consistent global and local explanations: SHAP provides like LIME local explanations but also a consistent global explanation like Feature Importance, since the Shapley values of each instance are the "atomic unit" of the global interpretations. Moreover, it maintains the feature dependence from the model trained.
- Robustness: SHAP always provides the same explanation for the same Tree Model, in contrast with LIME that includes randomness that makes the whole approach look unreliable.
- Reliability: The explanations obtained using SHAP are based on a solid theory and distribute the effects fairly based on the analysis of the original model trained. On the other hand, LIME surrogates the original model and, therefore, it can use features in the local interpretable model not used in the original model.
- Informativeness: The explanation from SHAP provides a very extensive explanation of how the model learnt, allowing the stakeholder and the scientist to be properly informed to support their decisions.
- Low computational cost: Although the computational cost of the

Shapley values are very high⁷, the computational cost for the tree SHAP is low (i.e., $O(TLD^2)$, T being the number of trees of the ensemble model, L the maximum number of leaves in any tree and D the maximal depth of any tree).

In the next section we will analyze both classification and regression from the Shapley values' perspective for the case of NTL detection. In the following Chapter 5 we offer a more in-depth analysis of each of the explanatory approaches used in this thesis.

4.3.2 Experiments: Classification vs. Regression Explainability in Real Data

Preliminaries

Data, Classification and Regression Algorithms For the experiments of this section, we use the classification and regression model from subsection 4.2.2 for the D_{A1} domain. Similar conclusions can be drawn for the rest of the domains.

Shapley Values and Interpretability To analyze the goodness of our model, we use the *summary-plot* method from SHAP. This method provides two plots for our type of problem (i.e., tabular data): a bar chart that represents the mean of each Shapley value of each feature, and a more complex plot that indicates how each value influenced (i.e., increased or decreased the prediction made from the base value). Both plots can be seen in Figure 4.4, applied on the classification approach. Regarding the second plot, in red there are the higher values of the features and, in blue, the lower values. When the feature is categorical there is no color scale and all the dots are gray. For example, in Figure 4.4 we can see that, on average, *Current Reading Absences* is the variable that contributes the most to the prediction, increasing the prediction when the value is high (i.e. the customer has had reading absences). In contrast, when there is no reading absences (i.e. *Current Reading Absences* = 0, in blue), the Shapley value is 0 or negative.

It is necessary to remark that when Shapley values correspond to the regression model, they can be read directly as the apportion to the standard output. In contrast, in the binary classification the Shapley value corresponds to the log odds ratio⁸. Moreover, it is necessary to clarify that the red/blue feature value representation is not valid for categorical features. In these cases, SHAP plots the dots in grey. Hence, Shapley

⁷ The Kernel SHAP cost is $O(TL2^M)$ in tree models, being M the number of features, T the number of trees and L the maximum number of leaves.

⁸ That is, x being the sum of the base value and the Shapley values from an instance, we would obtain the probability between 0 and 1 by doing $1/(1+\exp(-x))$.

values on regression have the additional characteristic of being simpler to interpret.

Considerations Regarding Subjectivity in the Analysis As it is widely analyzed in the literature, the supervised methods only detect correlations, hence human supervision is necessary to validate them as reliable causal patterns (or, at least, reliable correlation in the company’s context). For this reason, the following model comparison from Section 4.3.2 requires a human analysis of the Shapley values and therefore includes subjective considerations.

In general, a reliable pattern would consist of a correlation between a feature value x_i and the prediction \hat{y} that a stakeholder would trust. For instance, the stakeholders could easily validate patterns indicating that the customer is consuming less than expected based on their previous consumption or in comparison to other similar customers. A doubtful or questionable pattern would consist of those patterns that either cannot be easily validated by the stakeholders or whose interpretation is counter-intuitive (e.g. a correlation between a long period of average consumption and a high NTL score).

All these considerations are properly explained, in the following analysis, based on our experience in campaigns. In any case, we provide a fairly generic analysis that fits in most domains similar to the one used in this experiment. We try to avoid very complex analyses that could require information from the company (e.g. the historical NTL cases in specific towns) that cannot be disclosed.

Features Referenced in the Experiments The features referred to in this section are described in Table 4.2. For each model, we analyze in depth 8 features to ensure the readability of the document. However, we also provide a more generic description of the model that includes more information beyond the 8 features at the end of the analysis.

Evaluation Analysis through Explainability

According to Figure 4.4 and our interaction with the company’s technicians, we cannot trust the classification model since there is only one consumption-related feature in the top eight most important features (the *Min/Max bill last 12 Months*, a feature that refers to the ratio between the minimum and maximum consumption bill in the last year). Instead, many of the features are visit related (features that, as exemplified in Figure 4.2, can be useful but can also produce bias and other learning problems).

For a deeper analysis we can analyze the effect of each value on the output with the bottom plot from Figure 4.4:

- *Reliable patterns*: In the classification model, several patterns can be easily confirmed as true indicators of NTL:

FEATURE	DEFINITION
Current Reading Absences	After the installation of smart meters the company can remotely communicate with the meters. The absence of the meter readings can indicate either an incident in the meter (e.g. that it stopped working) or fraudulent manipulation. This feature indicates how many months have passed since the last meter reading.
Last Visit: Correct/Fraud	Categorical information that indicates if the last visit done to that customer has been correct or an NTL has been detected. If the customer has not received any visit, the feature's value is empty.
Town	The town where the customer lives.
# Meters in Property	How many meters the customer has in property. In general, the meter is owned by the company, and is rented/handed over to the customer.
Date Last Reading	How many months have passed since the last meter reading.
Last 'No Fraud' Visit	How many months have passed since the last time the customer was visited with a visit whose aim was not to detect fraud, i.e. the installation might not actually be checked during the installation.
Min/Max Bill Last 12 Months	The ratio between the minimum and maximum bill during the last 12 months.
Contracted Power	The contracted power by the customer. In general, it is expected that a customer with a high consumption needs a higher contracted power.
Cons. Zone/Cons. Last Year	Ratio between the consumption of the customer and the average in the zone from the same type of customers. A zone is an internal reference that refers to a group of customers that receive the electricity from the same point of supply. Therefore all the customers in the zone are very similar and have similar energy needs.
Last Bill	Last Bill in kWh.
Diff Consumption 6 Months	The difference in terms of kWh between two equivalent periods of time (i.e. the same six consecutive months) in consecutive years. A higher value indicates that the customer has had a consumption reduction.
# Months with No Consumption	Consecutive months with no consumption until the present.
Consumption Penultimate Year	Consumption of the customer in the penultimate year (i.e. from 24 to 12 months ago).

Table 4.2: Features referred to in the experiments with their descriptions.

1. *Current Reading Absences*: This feature is the most important feature for the model (according to SHAP). This is a very reliable pattern learnt because the company expects to have, after the introduction of smart meters, information from the meter on an ongoing basis, including meter readings. The lack of meter readings is for sure a very suspicious behaviour since it may indicate meter manipulation.
2. *Contracted Power*: According to the Shapley values there is a correlation between a higher contracted power and a higher probability of committing NTL. This pattern can be a bias since the company usually tends to include customers with higher Contracted Power in the campaigns. However, the company

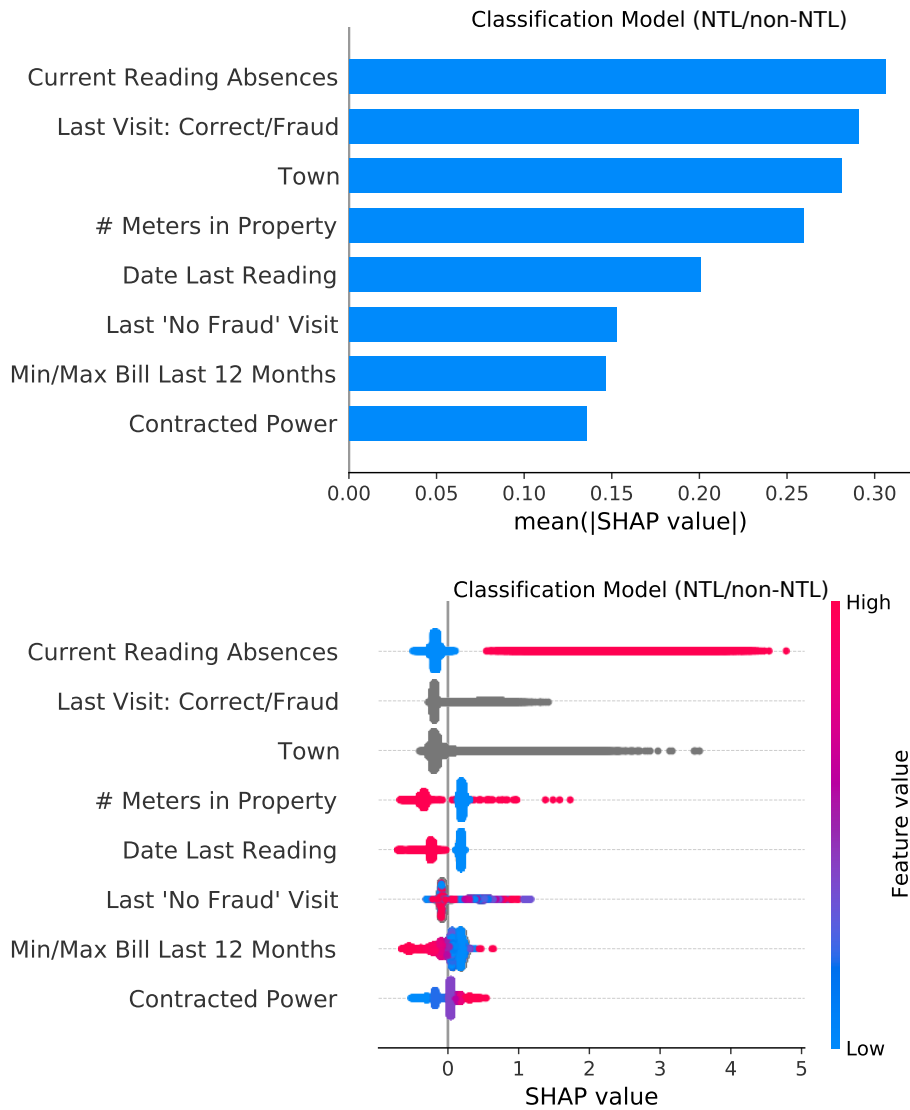


Figure 4.4: SHAP explanation of the classification approach: there is only one consumption-related feature on the top 8 most important features. Moreover, how each feature influenced in the score assignment is not easy to interpret: only the *Current Reading Absences* can be fully trusted as a good pattern and, for this reason, we cannot validate the model as a good and robust model.

validated this pattern based on their experience.

3. *Min/Max Bill Last 12 Months*: We can see that, in general, the model considers a lower value more related to NTL behaviour. We consider this pattern valid because, in general, we expect that monthly consumption will not vary in a very marked way

during the year. If this occurs, it may be a consequence of meter tampering.

- *Categorical information* Two categorical features (with no colour scale in Fig. 4.4 bottom) are very relevant in our system, as we explain as follows:
 1. *Last Visit: Correct/Fraud*: This information is valuable since the patterns learnt should be contextualised to the visits carried out by the company. That is, a customer that committed Fraud in the past is, according to the company, very likely to commit fraud in the future.
 2. *Town*: The town where the customer lives can be a good indicator for the NTL detection system. Statistically, there are towns in which the company has always detected more NTL cases than in other towns.
- *Unknown interpretability* The interpretation of how a feature value influences the output can be hard to understand for the classification approach. Several examples are given below:
 1. *# Meters in Property*: When a customer owns a meter, it is more likely to be in an inaccessible location. Therefore, it would be easier for the customer to manipulate it. Moreover, having more than one meter increases the possibilities of having an NTL. Therefore, one would expect that a high feature value would correspond to a high Shapley value. However, a high value in this feature influences unevenly on the output. With this information the stakeholder might not draw conclusions about the feature role in the prediction or its correctness.
 2. *Last 'No Fraud' Visit* Several interpretations can be expected for this feature. For instance, a recent visit combined with a high electricity consumption can confirm that a customer is not committing NTL, but also a recent visit to a customer that is consuming less than expected can be suspicious. The lack of context hampers the interpretation of the feature by the stakeholder.
- *Questionable pattern* Finally, there is a pattern learnt from a feature that the stakeholder cannot validate:
 1. *Date Last Reading* According to the SHAP value, low values (i.e. the last meter reading is recent) is more related to the NTL behavior. At first glance, this pattern is unintuitive since we would expect a similar pattern to the one learnt from the *Current Reading Absences*: a recent reading would indicate that the meter is working as expected. A possible explanation for this

unexpected output might be the correlation between the *Current Reading Absences* and the *Date Last Reading*: the model is already learning the expected pattern from the *Current Reading Absences*, and therefore the role of the *Date Last Reading* becomes unstable. Another option would be that the system is detecting an unexpected NTL pattern (e.g. a technician makes a manual meter read, detects an abnormal behavior and informs the company that the meter should be checked, and therefore there exists in the next few days another technician visit that confirms the NTL case).

Despite several aspects of the model being reliable in terms of NTL detection, the model relies on very few consumption features in the prediction process. This can be problematic in terms of robustness and fairness since the consumption features are better NTL predictors.

Instead, the regression model shown in Figure 4.5 is more robust, as it uses more consumption-related features, and it is easier to validate, as we explain as follows:

- *Reliable consumption patterns* In comparison to the classification model, the consumption features are the most relevant in the model:
 1. *Cons. Zone/Cons. Last Year*: Since we are comparing similar customers in terms of Tariff and region, we would expect that fraud corresponds to low consumption. This feature has learnt this pattern and, therefore, we consider it correct.
 2. *Diff Consumption 6 Months*: A high value indicates that in the past the customer consumed more than in the present. Therefore, the pattern learnt that a high value increases the output of the prediction and therefore should be considered reliable and correct.
 3. *# Months with No Consumption*: if the customer has several months with 0 kWh of consumption, it should be considered as a probable case of NTL, especially in populated regions and cities where there are not as many empty homes as in rural regions (at least in Spain).
 4. *Consumption Penultimate Year*: A high electricity consumption two years ago is not in itself a clear pattern of fraud. Nevertheless, it can be a very good complementary feature that indicates a change in consumption behavior. For instance, a customer who has always had low consumption is not the same as a customer who consumed in the past a lot and has recently changed their consumption behavior.
- *Reliable patterns from the binary model* Two important features in the classification model remain important in the regression model:

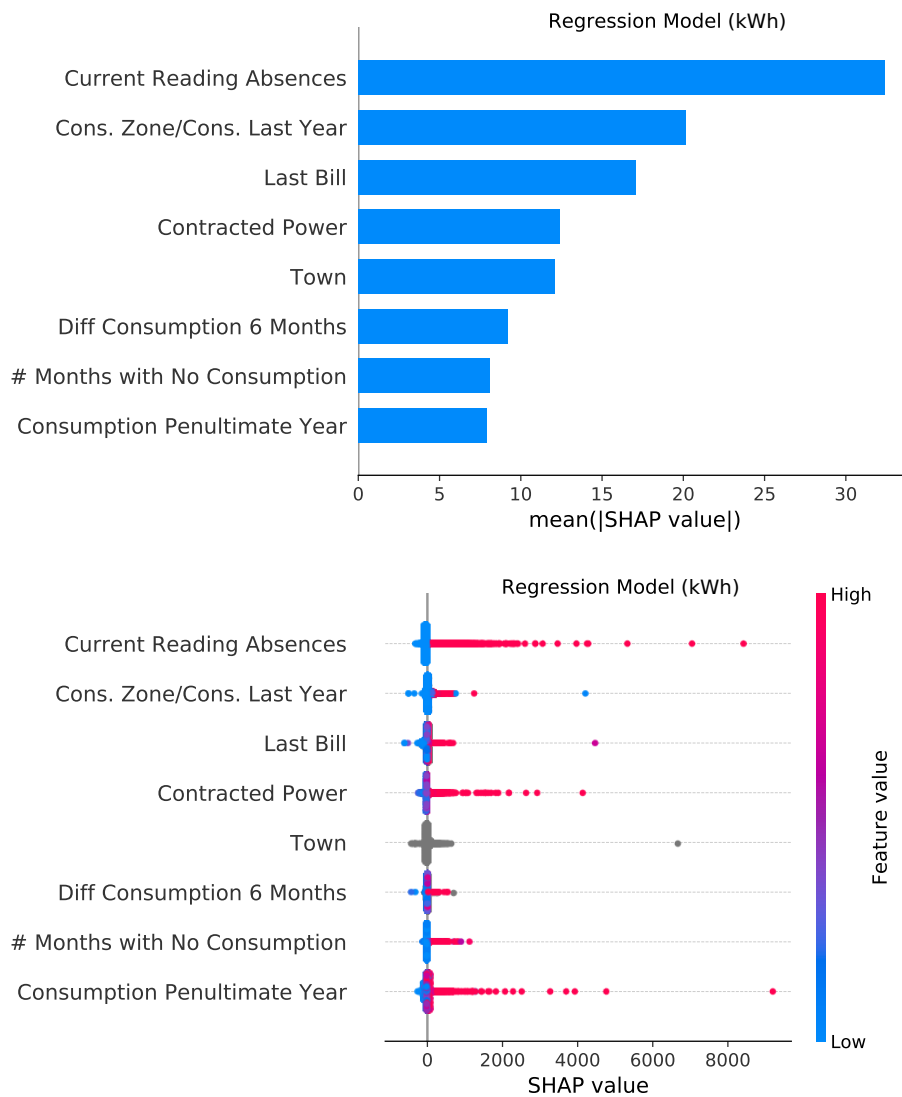


Figure 4.5: The regression model relies on consumption features to learn patterns and, therefore, we can consider that this model is better than the binary approach. Moreover, the patterns learnt seem to be easier to understand by the stakeholder, since more abnormal behaviors (the absence of meter readings or the number of months with no consumption) are more clearly related to a higher prediction than in the classification model, where lesser patterns can be easily trusted as trustworthy indicators of NTL.

1. *Current Reading Absences*: As explained in the previous analysis, the absence of meter readings is a likely indicator of NTL.
2. *Contracted Power*: The contracted power was also considered a very important feature in the classification approach. However, in the regression approach, the use of this feature makes more

sense: in the regression model we are trying to maximize the amount of energy to recover and, in general, the customer with a higher contracted power consumes more energy.

- *Categorical information* Only one categorical feature is in the top important features in the regression model:
 1. *The Town feature*: In comparison to the binary approach, the Town feature seems to have less relevance. However, we can see one specific Town value whose Shapley value is much higher than the other towns. This town corresponds to a small municipality where the company recovered a lot of energy in the past, and therefore it can be trusted.
- *Doubtful/Questionable pattern* Finally, we consider that there is one pattern in Figure 4.5 that the stakeholder cannot fully understand:
 1. *The Last Bill*: According to SHAP, a high value is learnt by the model as an indicator on NTL. The classical NTL behavior consists of manipulating the meter to avoid high bills and, therefore, we would expect the opposite behavior regarding this feature. However, there are circumstances in which a high last bill can be correlated with an NTL case:
 - A recidivist fraudulent customer that has been visited twice in a short period of time. The high bill corresponds to the back-payment of the previous fraud detected.
 - A customer with very high consumption that is not normal (e.g. illegal drug cultivation) that combines a correct installation of electricity with an illegal junction to get enough power.

In any case, these cases are more exceptional than the classic examples of reduced consumption and should therefore not be a pattern that is so prominent in the system.

This in-depth analysis of each model through their most important variables faithfully represents each model. For instance, the classification model only has 3 consumption features in the top 15 most important features, and 7 consumption features in the top 25 most important features according to Shapley values, while the regression approach has 10 and 19, respectively. In addition to that, it is tangible (as we have explained for each variable) that the patterns from the regression model are easier to analyze and corroborate by the stakeholder. This is true because as we have analyzed variable by variable, in the regression model, we can interpret what NTL patterns have been detected in that variable in a much simpler way. In classification, such analysis requires much more effort (the stakeholder cannot easily interpret what the pattern learnt by the model is), and the conclusions are often nuanced or unclear.

4.3.3 Customer Selection Through Local Explainability

Preliminaries: Local Explanation as Sanity Check

In Section 4.3.2 we have seen that the increase in energy recovered in Section 4.2.2 is justified because the regression model learns better patterns from the stakeholder’s perspective than the classification model. The resulting system is more robust since it learns less circumstantial patterns (e.g. fewer patterns related to the company’s decision that highly influence the observational data). Thus, the challenges regarding the lack of robustness and the low energy recovered per campaign generated are mitigated. Nonetheless, we can see in Table 4.1 that the system has room for improvement. That is, the system does not provide a perfect ordering of the customers according to NTL. Moreover, in Figure 4, we can detect that still, some non-NTL cases (or NTL cases with a very low amount of energy to recover) have a high score. In [31] we propose a solution to reduce the number of these undesired high-scoring customers with low or no NTL: to analyze through LIME the local explanation of each high-scoring customer included in the campaign, discarding those that, according to human knowledge, the explanation obtained is not reliable. Therefore, the final selection is a subset of the original sample.

In this section we propose an updated version using the local explanations of the Shapley values instead of LIME. This change of explanatory algorithm has two significant advantages. On the one hand, the Shapley values provide local explanations consistent with the global explanation of the model since the global explanation is constructed as the sum of the local explanations. On the other hand, the solid theory behind Shapley values (particularly the implementation for trees, i.e., tree SHAP) provides us with robust explanations (i.e. the explanations obtained for a model and prediction are always the same).

This sanity check has points in common with the analysis proposed in Section 4.3.2, where we analyze the correctness of the modular explanations. However, a good modular explanation does not guarantee that all the explanations at instance level of the top-scored customers are also reliable. Similarly, just because the model has learned a reliable and important fraudulent pattern at the modular level (e.g. a feature that, on average, greatly increases the prediction score) does not guarantee that all high-scoring customers have learned that pattern. Having said that, a good modular explanation, as it is built as the sum of the local explanations, should be an indicator of good explanations at instance level.

Post-Process Example

By way of illustration of this method, this example implements a simple rule system that automatically discards all the high-scored instances in which the most important fraudulent pattern (i.e. the feature value that increases more the prediction according to the Shapley values) is not

consumption-related. This is in line with the modular analysis from Section 4.3.2 in which we regard the regression model as a better predictor because the most important features are consumption-related.

This post-process approach aims to increase the campaign’s economic efficiency by increasing the amount of energy recovered per customer visited. Therefore, we compare in Table 4.3 the amount of energy recovered for each customer on average in an n -sized campaign⁹, for the Domain D_{AN} . As expected, we can see in Table 4.3 that the regression approach outperforms the classification approach in terms of energy recovered per customer visited. However, our post-processing at instance level implemented in the regression approach outperforms the regression approach by up to 34%.

Average energy recovered per customer in an n -sized campaign (kWh)				
Domain D_{AN}	$n = 528$	$n = 211$	$n = 106$	$n = 42$
Reference	2107	3782.9	5495.1	8716.4
Classification	823	930.8	921.3	900.9
Regression	887.3	1266	1554.8	1740.3
Regression+Rule	944 (+6%)	1398.4 (+10%)	1741.5 (+12%)	2328.7 (+34%)

Table 4.3: The post-processing at instance level (by not including those customers whose most important fraudulent feature according to the Shapley values is not a consumption-related feature), referred to in the table as *Regression + Rule*) reduces the size of the selection but increases the amount of energy to recover on average for each visit. More specifically, 31 out of 42 customers, 84 out of 106, 173 out of 211 and 469 out of 528 customers would be included in the final campaigns, but in each case, we would increase the amount of energy recovered per customer visited, a clear indicator that this post-process would discard more non-NTL cases (or NTL cases with low energy recovered) than otherwise. That is, we increase the economic efficiency of our campaign, recovering more energy per visit carried out by the technician.

In this example, we have used a straightforward rule to provide a rather generic example. However, this approach is very useful to nuance the campaign based on the stakeholder’s knowledge. For instance, as we explained in Section 4.2, one of the existing biases is related to the fact that the company generates campaign to over-control historically fraudulent customers. From our perspective, this pattern is valuable and trustworthy since many fraudulent customers are recidivists. However, we would like to avoid high-scoring customers with only this pattern as an indicator of NTL. Therefore, this post-process method would be helpful to discard these specific high-scoring customers that would not be humanly validated.

⁹ n corresponds to the customers preselected for the campaign, as explained in Table 4.1.

4.4 Discussion

This chapter introduces the NTL detection system grounded on regression as a valid alternative to using classification. Moreover, we illustrate the use of explanatory algorithms to understand the predictions of the system. Experiments performed indicate that using the energy recovered as the priority setter helps the system be more successful, mitigating the biases problems regarding the use of observational data. The patterns learnt are easier to validate from a human perspective, and therefore the models generalize better. Surprisingly, the use of regression in the NTL literature is scarce. For instance, [76] describes an outlier detection system, where the amount of energy to be spent by a customer is forecast. We believe our approach can be enhanced by using the techniques in the aforementioned work.

On the other hand, this work is one of the few examples in the literature that implements explanatory algorithms for NTL detection. Our experiences and lessons learned can be useful not only for any initiative that aims at increasing interpretability but also for any data-oriented industrial project. The following chapter delves deeper into the subject of interpretability, explaining the multiple methods used in this project.

Chapter 5

Introducing the Human Perspective

5.1 What is my System Learning?

Our first approach to building the NTL detection system (Chapter 3) achieved good results, but lacked robustness and transparency. For this reason, we started to analyse in depth what our model was learning through statistical methods such as explanatory algorithms. The initial result of this analysis can be seen in Chapter 4, where through Shapley values we were able to better understand what our model was learning at a modular and local level, and we were able to propose key improvements in our system, such as the use of recovered energy as a variable to be predicted instead of using the binary information NTL/non-TNL. However, we do not provide details regarding why we use Shapley values instead of other existing alternatives in the literature.

This chapter details our experience of providing the desired transparency in our NTL detection system. We explain how we started to use simple statistical analysis (i.e., odds-ratio, Pearson Correlation and Feature Distribution), then how we used the feature importance implementation from the ensemble tree methods, to finally using state-of-the-art explainability methods like LIME (for tabular data) and SHAP (tree explainer). Finally, we compare each approach in our context¹, concluding that we consider Shapley values the best approach in our context and, in many cases, the best approach seen in the literature to provide explainability in an industrial project.

The analysis and tests carried out in this chapter include several conclusions already referred to previously in Chapters 3 and 4. For instance,

¹ The labelled information used in this analysis corresponds to the dataset used in [30], a rich dataset with almost three hundred three thousand labelled instances, containing 3.3% of NTL cases. The model M used to make the predictions is a CatBoost regression model tuned through a training-validation process.

the models used to compare the explanations from the explanatory algorithms are regressive, since as we explained in Chapter 4 the regression approach gives us better results than the classification one.

Features	Definition
Last Impossible 2	Indicates the number of months since the last time the company could not perform a visit. The suffix 2 indicates that the original purpose of the visit was not to detect an NTL case, but to implement another technical visit.
# Impossible 2	Number of times the company could not perform a visit to the labelled customer. The purpose of the visit was not to detect NTL.
# Correct 2	Number of “no Fraud” visits to the customer with result of “no NTL detected”.
Last Impossible	Number of elapsed months since the last time the company could not perform a visit. We compute both the visits to detect NTL cases but also the other visits (referred in this analysis, as explained before, with the suffix 2).
Current Reading Absences	Number of months since the last meter reading.
Power Contracted Town	The power contracted by the customer.
Town	Town where the customer lives.
# Visits 2	Number of “no Fraud” visits made to customer. We do not consider the results of the visit, but if the customer has been historically “controlled” by the company.
SCC12MP	Similarity (consumption curve, 12 months) between the customer and similar customers. We compute the average consumption per month of the customers from the same province and Tariff, normalize the consumption curve, and compute how similar this normalized consumption curve is to the normalized consumption curve of the customer. A low value would indicate that the consumption is similar to the expected consumption curve, while a very high value indicates that there is no similarity (i.e., an indicator of NTL).
SCC12M	Similarity (consumption curve, 12 months) between the customer and similar customers. This variable is computed as explained in SCC12MP , but the comparison is done with all the customers included in the campaign.
Con.Penultimate Year	Consumption of the customer during the penultimate year. This information might not provide meaningful data, but it is useful to understand the historical consumption behaviour of the customer and, therefore, can nuance the meaning of consumption behaviours at the present time.
Last Bill	Amount of energy billed to the customer in the last bill.
Last Fraud 2	Number of months since the last time the company performed a “no Fraud” visit with an NTL result.
Con.Drop Abs 24-6M	Absolute consumption difference (i.e., kWh) between two consecutive 6 month periods. It is checked for the last 24 months.
Last Threat	Number of months since the last time the customer threatened a technician from the company preventing the meter’s technical service.
#Threat	Number of times the customer threatened a technician from the company preventing the meter’s technical service.
Energy Cut	Number of times the company cut the energy supply to the customer.
Fraud Building 1Y	Number of times an NTL has been found in the customer’s building during the last year. This information is interesting since neighbors can share information about how to commit fraud.
#Gas Fraud	Number of times the customer has had NTL cases in gas.
Con.Last Year/- CLY Zone	Ratio between the customer’s consumption and the average consumption in the region (last 12 months). This information is straightforward, i.e., a much lower consumption should be an indicator of NTL.
Con.Low 24-6M.	Ratio between the customer’s consumption and the average consumption of similar customers (i.e., customers from the same region and Tariff). The period of time considered is the last 24 months, and the consumption window is 6 months.
Con. Customer/- Con. Cust. 24M	Ratio between the customer’s consumption and the average consumption in the region (last 24 months).
Median Bill 12M	Median Bill of the customer for the last 12 months.

Table 5.1: Brief description of the features referred to in this chapter.

5.2 Our Experience Using Explanatory Algorithms

5.2.1 The Starting Point: Statistical Analysis

Our first approach to understanding our predictive models was to analyse the training dataset statistically. The statistics measures referred to in this work are the following:

Feature Distribution The distribution of the values of each feature in different domains and segments might indicate patterns and other interesting characteristics to analyse. For instance, this simple analysis might provide a small intuition on why, in certain regions, the company has been more successful in detecting NTL cases.

Pearson Correlation A measure of the linear correlation between two features, where 1 indicates a perfect positive correlation (i.e., for every increase in one feature, there is a positive increase of a fixed proportion in the other feature), and -1 indicates a perfect negative correlation (i.e., for every increase in one feature, there is a decrease of a fixed proportion in the other feature), where 0 indicates no linear relationship. The coefficient (r) is defined as the ratio between the covariance Cov of the values of two features divided by the product of their standard deviation S , i.e.:

$$-1 \leq r_{XY} = \frac{Cov(X, Y)}{S_X S_Y} \leq 1 \quad (5.1)$$

There exist in the literature examples of using the Pearson Correlation Coefficient to detect NTL. For instance, in [88], the Pearson Correlation coefficient is used to detect an abrupt and gradual but constant decrease of consumption in customers, hence suspicious of NTL. Thus, the Pearson Correlation can be used to detect patterns in our data useful in understanding NTL behaviors.

Odds-Ratio The Odds-Ratio OR statistic is usually used in medical reports (as explained in [16]). It quantifies the influence of a binary value on an outcome. In an NTL detection context, let $F_{x_i=1}$ be the number of NTL instances x with feature $x_i = 1$, $F_{x_i=0}$ be the NTL instances x with feature $x_i = 0$, $C_{x_i=1}$ be the non-NTL instances x with feature $x_i = 1$, and $C_{x_i=0}$ be the non-NTL instances x with feature $x_i = 0$; then the OR is:

$$OR = \frac{F_{x_i=1}/C_{x_i=1}}{F_{x_i=0}/C_{x_i=0}} = \frac{F_{x_i=1}/F_{x_i=0}}{C_{x_i=1}/C_{x_i=0}}. \quad (5.2)$$

Odds-Ratio values far from 1 indicate that customers with $x_i = 1$ and customers with $x_i = 0$ have a different proportion of NTL.

These statistical metrics are not often considered explanation methods, but help understand the data used to train the model and, in some cases, are sufficient to detect biases or undesired prediction rules. In part,

many of the problems explained in Chapter 3 were detected using these statistical metrics (for instance, the over-representation of the recidivist customers explained in Chapter 4). Nevertheless, the statistical analysis could not provide satisfactory explanations for how our black-box model made predictions. For this reason, we started to introduce different explanatory approaches in our system to better understand the role of each feature in the prediction process.

5.2.2 Feature Importance

The Feature Importance method, as explained in Section 2.3.2, provides a simple modular explanation of how important each feature has been during the training process. In this case, we will analyse the feature importance method in Catboost, more specifically the *PredictionValuesChange* method that evaluates how much the prediction changes on average if the value of that feature is changed². The result of the method is a ranking of features by importance, where importance values are normalized so that their sum equals 100.

Figure 5.1 shows the top 10 features³ of our *M* reference model. This information provides a global picture of the learnt model, since the most important features to detect NTL cases are consumption features (and, to a lesser extent, visit features). Overall, the fact that 8 out of 10 features are related to consumption or visits (and not, for example, to the town where the customer lives) convinces the domain experts that the model focuses on the right information. Hence, using feature importance is helpful as a first sanity check of the model.

However, feature importance is insufficient when it comes to analyzing in depth how features influence the prediction. See for instance the most important feature, *Last Impossible 2* in Figure 5.1; this feature indicates the last time the company failed to carry out a “No Fraud” visit. A “No Fraud” visit is one whose main aim is not to detect fraud, but some other generic purpose. Yet, the impossibility to perform the visit may hide an abnormal customer behavior (e.g., the customer obstructs the technician’s visit because they know the meter has been tampered with). Although the model has learned this fraudulent pattern, it cannot be confirmed through feature importance, since it only provides a global, fast⁴ explanation of the importance of the features in the model, but does not provide the reason behind a high relevance. Indeed, in the ideal case, a high relevance describes a learned pattern of NTL. But it can also be the result of a bias

² The definition of the *PredictionValuesChange* is available in the documentation from CatBoost, https://catboost.ai/docs/concepts/fstr.html#fstr_regular-feature-importance

³ In this section we will only analyze the top *n* features of each method to facilitate the explanation.

⁴ In our tests with a catboost model with 3056 trees and 605076 instances takes less than 0.05 seconds to compute the Feature Importance. The hardware used in all the tests is an Intel Core i7-8550U CPU, with 16GB of RAM and an SSD disk.

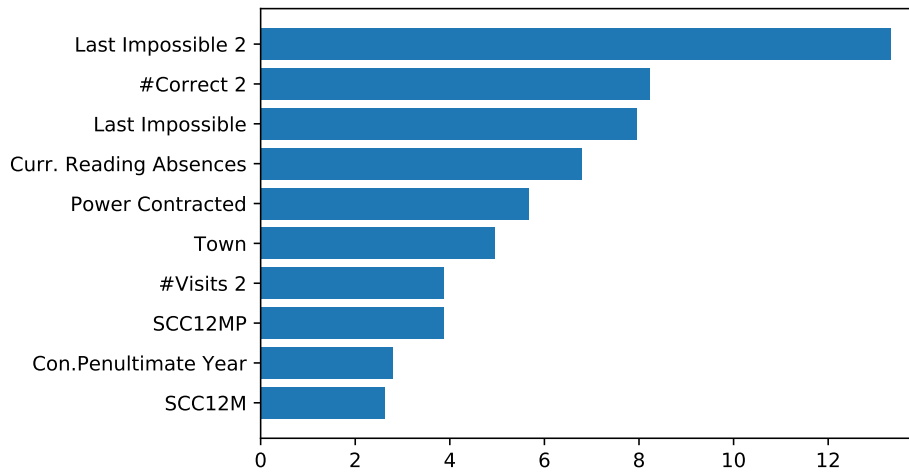


Figure 5.1: Top 10 most important features according to the feature importance method from Catboost, i.e., PredictionValuesChange. It evaluates how much the prediction changes if the value of that feature is changed, on average.

in the data, or an internal decision of the learning algorithm that is not always justified or understandable by the stakeholder.

In this situation, it is necessary to complement the Feature Analysis with, for instance, a statistical analysis. In Table 5.2 we analyze the distribution of the *Last Impossible 2* feature. In general, most of the labelled instances are undefined for this feature (i.e., there are no cases of *Last Impossible 2*), but this proportion is reduced when we focus our analysis on the NTL cases with more than 3500kWh⁵ recovered (where more than 10% of the cases had a *Last Impossible 2*), and the ratio is reversed for the very top cases, where 75% of the customers had a value for *Last Impossible 2*. This pattern should be the one learned by our model.

In conclusion, based on our experience, the feature importance methods might not be a proper method to fully understand the patterns learned by the NTL detection model, but can be a good baseline approach to detect clear undesired patterns.

5.2.3 Local Surrogate Models

This section discusses an example of the explanations obtained from LIME for tabular data. We fix one particular customer for the remainder of the section⁶. This is an NTL case for which a fraud of 3000kWh was

⁵ The consumption of a house or apartment in Spain is, on average, around 3500kWh. Therefore, this figure is informally used in the project as a delimiter of what would be a great NTL case, i.e., an NTL case in which the amount of energy recovered is remarkable.

⁶ The execution time to obtain an instance explanation for LIME is, in our case, around 38 seconds.

Top Selection	Last Impossible 2 undef.	Last Impossible 2 >=0
3500 to 35000kWh	571	84
>35000kWh	1	3
Customers	295218	7324

Table 5.2: Analysis of the value *Last Impossible 2* feature for the NTL labelled customers with recovered energy from 3500kWh to 35000kWh, more than 3500kWh and all the labelled customers. This feature indicates the months passed since the last *Last Impossible 2* visit, where the value remains undefined (i.e., the missing values) in case of no visit. The proportion of *Last Impossible 2* ≥ 0 increases for the NTL cases, especially when the energy recovered is high.

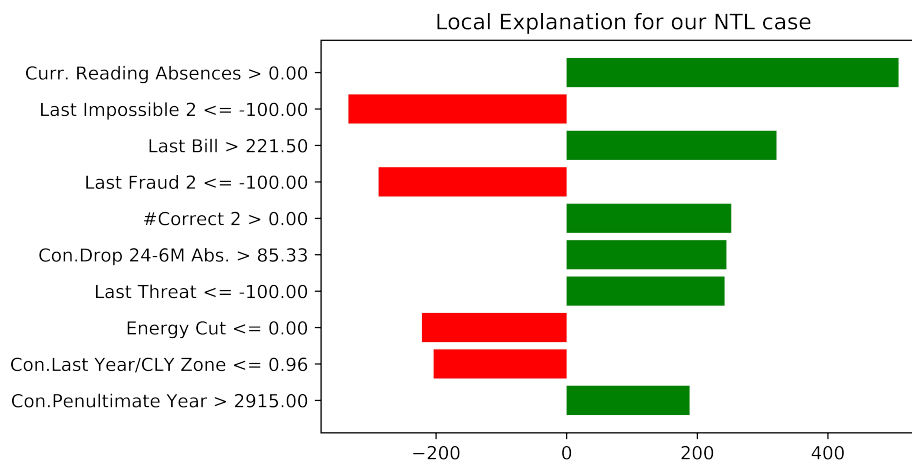


Figure 5.2: Local Explanation of the NTL case, first run. The top-10 most important features from the LIME explanation of an NTL case with energy recovered of around 3000kWh. The most important features are the *Current Reading Absences* (i.e., that it has absences in readings) as an indicator of NTL and the *Last Impossible 2* feature (i.e., that has a negative value, indicating the absence of Impossible “No Fraud” visits as a non-NTL pattern).

reported; our model predicted an amount of energy to recover of around 2100kWh. Note that all features discussed are numerical, since LIME requires re-encoding categorical variables as numerical ones, as explained in the documentation⁷. This example has been carefully selected because it exemplifies the problems we have had with LIME in our system.

Figure 5.2 shows an example of a subset (top-10) of the most important features for that customer according to LIME: the explanation indicates which feature values increase or decrease the specific prediction of

⁷ <https://marcotcr.github.io/lime/tutorials/Tutorial%20-%20continuous%20and%20categorical%20features.html>

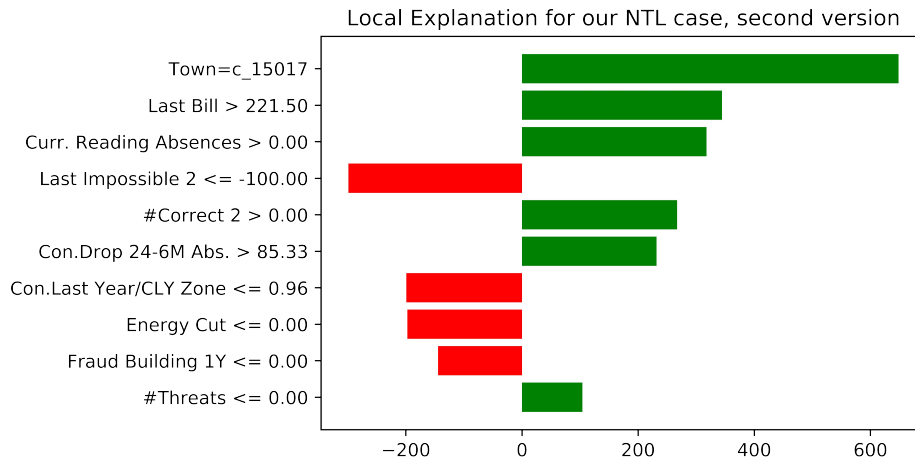


Figure 5.3: Local Explanation of the NTL case, second run. The same instance of Figure 5.2 is explained differently by LIME in a second run, due to sampling a different set of neighbours.

2100kWh. The sum of each prediction apportion should be the prediction made by the black-box model in that local region (or at least, a good approximation).

However, LIME seems to have a major robustness problem, exemplified in Figures 5.2-5.3. If the LIME algorithm is rerun on the same instance, a different random sample is generated each time to generate the local model, and this leads to different explanations of the same instance.

A second issue with LIME, reported elsewhere, is the high sensitivity of the output to the setting of certain parameters, particularly kernel width. For instance, in Figure 5.4 we show the explanation of the same instance of Figure 5.2, but now using a different *kernel_width* value. There is also little theoretical guidance for choosing appropriate values.

Finally, there is no guarantee that the explanation that we obtain from the local surrogate model is faithful to what $M(x)$ computes, as shown in Figure 5.3. The *#Threats* feature indicates if the company’s technician has received threats from the customer when performing an installation or service, and *Energy Cut* indicates whether energy has been cut off to this user at some point in the past. However, upon closer inspection, these features are not used in $M(x)$ computation.

Despite these problems, based on the information provided by LIME, a methodology can be proposed. In [31], we describe an approach for double-checking the predictions made by a model by implementing a rule system. Based on the features that, according to LIME, most influenced the score for each instance, this methodology would determine if the high prediction was trustworthy, discarding as NTL-cases those instances for which, according to human knowledge, a high score is not justified. The accuracy in our tests increased around 13% with this simple heuristic, as

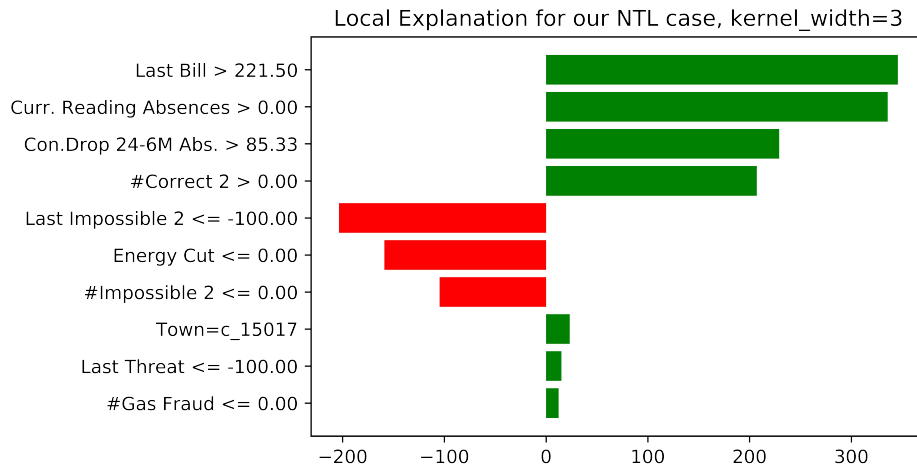


Figure 5.4: Example of how the *kernel_width* highly influences the explanation process.

shown in Table 5.3.

Dataset	% NTL	% Non-NTL	% Precision
Original Campaign	72	28	72
LIME Campaign	18	3	85

Table 5.3: Results from the tests in [31] where we used LIME as a post-process method to rule out customers with an unjustifiably high score.

5.2.4 SHAP

As explained in subsection 2.1.4, we consider it mandatory to analyze the goodness of a model by analyzing how it has learned to score: if we only focus our analysis by comparing experiment benchmarking in a validation test dataset, we might perceive as good models those that have learned undesired patterns and, therefore, might fail in the real-world scenario. In this work we propose using the SHAP explainer method that, as introduced in Section 4.3, explains how the supervised model predicted \hat{Y} . The overall process of how the Shapley values work is explained in Section 2.3.2; here we only include a reminder of how the values are interpreted:

1. The supervised model predicts \hat{Y} .
2. Based on these predictions, SHAP extracts the Shapley values, and the base value.
 - Base Value: The mean of the labelled instances the supervised model used in the training stage.
 - Shapley Values: Each value of each instance has an associated Shapley value, that corresponds to the influence of said value

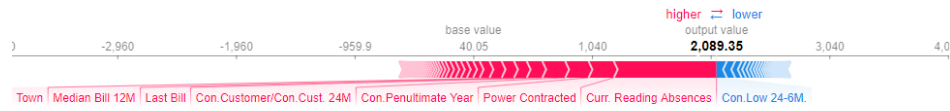


Figure 5.5: Example of the Shapley values for our reference NTL case of 3000kWh. The most important feature for this instance is the *Curr. Reading Absences* and the *Power Contracted*.

to the final prediction (i.e., if that value pushes the final score, increasing or decreasing it).

3. The final prediction for each instance can be understood as the result of the sum of all the Shapley values of that instance and the Base Value⁸.

We will use the bar chart that represents the median of each Shapley value for each feature and the fuller version of it that indicates how each value influenced (i.e., increased or decreased the prediction made).

In this section we analyze the explanations from SHAP for trees (i.e., the Tree Explainer). We use the same reference instance used in the previous section, i.e., the positive NTL customer for which 3000kWh of fraud was reported and for which our model M predicted 2100kWh of recoverable energy.

Figure 4.4 shows the explanation of SHAP of a subset (top-10) of the most important features for our reference instance. Similar to LIME, SHAP indicates how the feature values increase or decrease the specific prediction of the energy to be recovered. Furthermore, it does not have the robustness problems of LIME since the computation is deterministic and always provides the same explanation for a given model and instance⁹. In addition, the explanation is consistent with what the model has learned, which was not always the case in LIME as described before with the *Energy Cut* feature.

SHAP for tree-based models [81] is, according to our experience, a very robust and rich method to provide interpretability to our system. To begin with, the fast implementation¹⁰ provides local explanations (i.e., instance level, see Figure 5.5) and global explanations (i.e., modular explanations, similar to the feature importance method previously analyzed, see Figure

⁸ It is necessary to remark that SHAP, at least in the XGBoost and Catboost binary models that we have tested, the Shapley values do not correspond to the direct probability but a raw value from the learning process. In our case, to obtain the real probability from the Catboost model, it is necessary to calculate the sigmoid of that value, i.e. $\frac{1}{1 + \exp(-value)}$

⁹ For the Tree Explainer used in our system.

¹⁰ In our case, the system computed the Shapley values in around 260 seconds. We did not use the GPU accelerated version of the Tree Explainer, which for sure would provide an even faster computation but would require an Nvidia GPU.

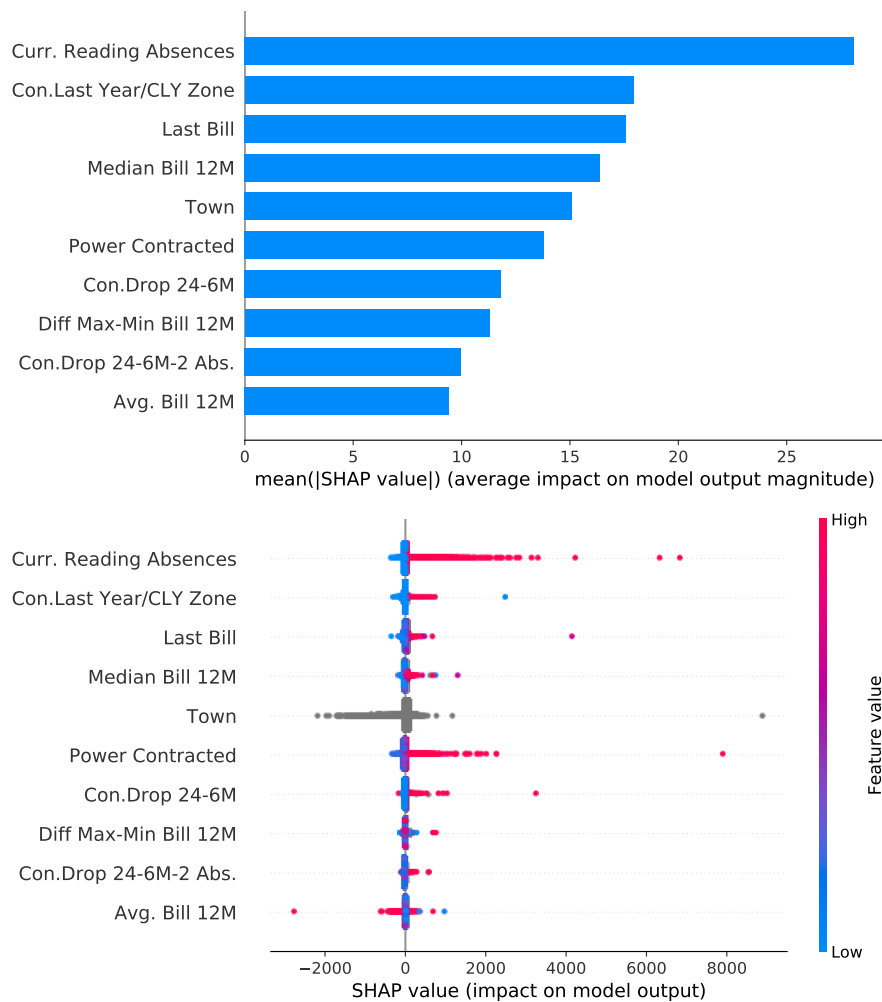


Figure 5.6: Two versions of the *summary_plot* from SHAP: above, a box-plot diagram showing the average of the Shapley values for the most important features (similar to feature importance plots). Below, a distribution-like description (more fine-grained) of the same information. According to SHAP, the most important feature in the model is *Curr. Reading Absences*.

5.6). Remarkably, this global explanation is consistent with the local explanations (as explained in [87], “the Shapley values are the ‘atomic unit’ of the global interpretations”), and with the feature dependency of the predictive tree Model. Moreover, the theory of Shapley values [121] guarantees the properties of *efficiency* (the feature contribution adds up the difference of prediction), *symmetry* (two features have the same Shapley values if they contributed equally), *dummy* (a feature that is not used in the prediction model has a Shapley value of 0) and *additivity* (if the computation of the prediction can be divided into sub-processes, i.e. the boosting process in our model, the Shapley values can be seen as the average of the Shapley

values of each tree).

5.2.5 Comparison in our NTL Detection System

	Statistical Analysis	Feature Importance	LIME (tabular data)	SHAP (tree SHAP)
Coverage	Data	Model	Model	Model
Model Agnostic	Yes	No	Yes	Yes
Robustness	Stable	Stable	Unstable	Stable
Scope	Global	Global	Local	Local/Global

Table 5.4: Comparison in terms of Coverage (what the method analyses), Model-Agnosticism (if the method can be used in any model), Robustness (if the methods always provide the same explanation for the same data) and Scope (if the method explains an instance or provides a global explanation of the model) between the four methods considered.

Feature importance, the local approach from LIME for tabular data, and the SHAP method for tree-based trees, are three methods that represent a step forward to better understand our predictive models in comparison to the statistical methods referred to in Section 5.2.1. Each method offers a different approach to the goal of explaining the models, with its advantages and disadvantages, which are discussed below:

- **Depth:** The big difference in terms of depth is that Feature Importance provides a superficial and modular explanation of the influence of each variable on the predictions, while LIME and SHAP offer deeper explanations at the instance level. Therefore, as we have seen in our use case, Feature Importance can be interesting because of its speed in getting a first sanity check of the model, but its superficiality would not allow us to implement the double-checking methods exemplified in [31] (that uses LIME) or [35] (that uses Shapley values).
- **Bias Detection:** Feature significance is a good approach to easily detect biases and other data-related problems. However, this can also be done with SHAP, which by complementing such information with local explanations, gives us a better insight into which values cause biases. In contrast, LIME’s local approach makes it much more complicated when it comes to analyzing biases and unwanted patterns in the model due to the lack of global analysis.
- **Robustness:** LIME has the problems of robustness of explanations across runs due to its random component, which makes the whole approach look unreliable. In contrast, feature importance and SHAP (Tree SHAP) always give the same results for the same data.
- **Truthfulness:** Feature importance and SHAP compute the importance of the features by analyzing (with very different approaches

and with a different focus) how the prediction changes when there is a modification in the feature. The local model from LIME, on the other hand, can use features in the local explanation not used by the model (and therefore the explanation is not trustworthy).

- **Complexity:** Obtaining explanations for each method, in our case study, is fast:
 - The Feature Importance provides a superficial modular explanation in much less than a second.
 - The LIME method provides a local explanation in around 38 seconds.
 - SHAP provides local and global explanation in around 250 seconds.

Thus, we could conclude that no approach can be discarded because it is computationally expensive. That said, it is worth noting that LIME offers local explanations (i.e., if we wanted a global explanation of the system, e.g., which variable might be relevant in general, we would have to compute the explanations multiple times). Regarding SHAP, we should also take into account the computational cost of obtaining the explanations since we use the implementation specific for tree-based models¹¹.

Thus, SHAP is the method we consider the best of the three methods tested to explain our models, and the one we have worked with to analyze our system. With SHAP we have local and global explanations consistent with each other in a fast way, with the guarantee that the explanations are robust, always obtaining the same explanations for the same model and dataset. This gives us confidence in the explanations obtained, allowing us the data scientists, to make decisions on improving the system based on the explanations obtained, as well as giving stakeholders confidence in their model and explanations, thus acquiring useful new knowledge.

5.3 Discussion

As explained in previous chapters, developing a robust NTL detection system is challenging. For this reason, many of our efforts during its development has involved better understanding our system, i.e., achieving transparency in our predictions to mitigate and correct many of the flaws in our system. In this chapter we provide an in-depth explanation of all the

¹¹ TreeSHAP has a computational cost of $O(TLD^2)$, being T is the number of trees, L the maximum number of leaves in any tree and D the maximal depth of any tree, while the KernelSHAP (the model-agnostic approach that can be used for any type of algorithms such as neural networks, support vector machines or tree-based models) cost is $O(TL2^M)$ in tree models, being M the number of features.

methods used to achieve this goal, analyzing their advantages and disadvantages, concluding that the method that gives us the best explanation, both globally and locally is SHAP's Shapley values.

In the next chapter we take interpretability considerations a step further. More specifically, we analyze the role that the stakeholder should have in the NTL detection system. We propose a human-in-the-loop system to empower stakeholders so that they can have full control of the system and the campaigns to be generated.

Chapter 6

Human-in-the-Loop Approach to Improve NTL Detection

6.1 Mitigating the Existing Problems for each Model Built

In previous Chapter 5 we have analyzed how explanatory algorithms, and more specifically Shapley values, can provide us robust explanations that allow us to validate the patterns learnt by the model, as well as compare beyond benchmarking different approaches with the aim of analyzing which one would generalize better on unseen data. This allowed us to implement different solutions that would improve our system overall. However, as we explain in our work [34], there exist several biases and other data related problems that cannot be treated with a general approach, since it affects specific domains. In other words, the observational data causes different specific biases in each domain that it is difficult to treat at once.

Thus, in this Chapter we present our step forward to exploit the information provided by the Shapley values: to convert the process of building our model into a human-in-the-loop process controlled by the stakeholder in charge of the NTL detection process. In each iteration, this specialist analyses what the model has learned and implements feature engineering to improve the model if it detects an undesired pattern, a bias, or an unused feature. After several iterations, as we exemplify in this work, the resulting model is better in terms of accuracy, robustness, interpretability, generalizability, flexibility, and simplicity.

6.2 Human-in-the-Middle Approach to Implement Specific Solutions for each Domain

6.2.1 The Proposal

In this chapter we propose to involve the stakeholder through a human-in-the-loop solution to guide the system when training the model (Figure 6.1). In each iteration, the stakeholder analyzes through Shapley values the patterns learned and implements feature engineering to correct biases and other data-related problems that are specific to that domain at the moment of building the campaign, as well as remove correlated or unused features to increase the system’s interpretability, to achieve a simpler (i.e. with fewer variables), more understandable (i.e. with patterns validated by the stakeholder) and, therefore, a better model in terms of generalization.

To benchmark each model we use the Normalized Discounted Cumulative Gain (*NDCG*, [70]) to obtain a global vision of the quality of the predictions made by a model,

$$NDCG_t = \frac{DCG_t}{iDCG_t}$$

where DCG_t is defined as

$$DCG_t = \frac{\sum_{i=1}^t energy_i - 1}{\log_2(i + 1)}$$

being $energy_i$ the amount of energy recovered in the visit made to the customer ranked at position i , and $iDCG$ corresponds to the maximum DCG possible (i.e. a perfect prediction in terms of order). The *NDCG* provides a generic vision of the correction of the model beyond any threshold. Other alternatives (e.g. *precision@k*, i.e. precision at the top k instances) can tend the model to exploit the existing biases in the data and, therefore, the resulting model would not be generalizable.

6.2.2 The Human Analysis in the Building Process

With the information provided by the Shapley values and the *NDCG* metric, the stakeholder has to analyze in each iteration n the correctness of the model trained in comparison to the previous iteration $n - 1$, and propose a new model in iteration $n + 1$ by implementing feature engineering. This process is subjective (e.g. depending on the stakeholder there might be slight differences regarding what can be considered a good pattern), and also needs to be adapted to every domain (e.g. a good pattern in one domain might be a fair pattern in another domain). However, there are certain fundamentals that every analysis shares, summarized as follows, which we will use in this work:

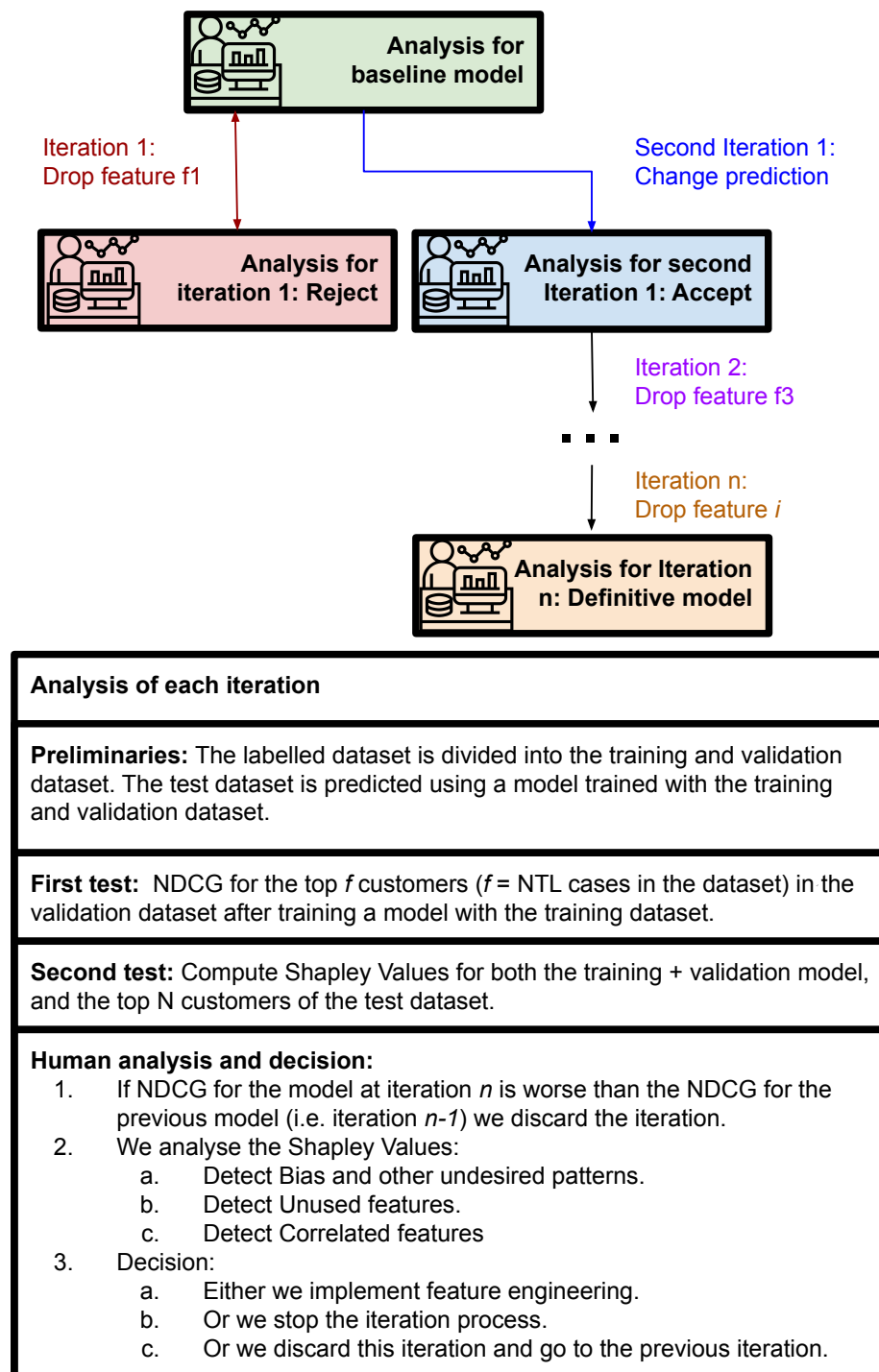


Figure 6.1: The building process is an iterative human-in-the-loop process where the stakeholder uses Shapley values to guide the model's training process to achieve a more generalizable and fairer model.

The *NDCG* Should not Decrease between Iterations In each iteration, we should not decrease the benchmarking performance on unseen data. Therefore, in each iteration, $NDCG_n \geq NDCG_{n-1}$ ¹.

The Outliers Should be Detected and Processed A Shapley value from a high-scored instance that stands out in comparison to the rest of the Shapley values can be a consequence of an outlier in the prediction labels (more specifically, an NTL case with a much higher value of kWh recovered than the rest of the NTL cases). In this case, the stakeholder should consider transforming the instance that causes the outlier to avoid biases on prediction.

The System Should be as Simple as Possible To reduce the complexity of the model to increase generalizability on unseen data but also improve interpretability, we should remove features that have a low impact on the model. Also, we should remove correlated features with similar meanings that contribute similarly according to Shapley values.

The correction of bias should have priority over removing a feature: a bias highly influences how a model is learned and, therefore, its correction can cause a feature with no importance in the biased model to gain relevance in the new model. All these considerations are explained in the short example that we provide in the following Section 6.3.

6.3 A case Study with a Real Dataset

In this section, we exemplify the human-in-the-loop process and analyze the benefits of implementing it in our NTL system.

6.3.1 Preliminaries

The Dataset Used

For the case study we use a real dataset² from the utility company with more than 1,000,000 customers³. The labelled instances include around 10,500 NTL cases, and almost 300,000 non-NTL cases and the dataset is split into three sub-datasets: the training (80% of the labelled instances), the validation (10% of the instances) and the test dataset (the remaining 10%). Each partition is stratified. There is no timestamp consideration

¹ We would accept some margin in this description, i.e. we consider that a model is worse in terms of *NDCG* when the value is significantly lower (at least 0.1 lower).

² further information like the region and the typology of the customers is anonymized to protect the privacy of the data.

³ The customers are apartments/small houses from the same Spanish region.

(i.e. we do not use the last 10% of NTL cases as the test dataset) to guarantee diversity and reduce the differences between the datasets⁴.

The Algorithm, Loss Function and Metric Used

The Gradient Boosting Model trained is a Root Mean Square Error Catboost Regressor, i.e. we consider the problem of detecting NTL as a point-wise ranking problem where we predict the amount of energy to recover for each customer. The methods used to analyze the correctness of our model are the *energy*₂₀₀ (to compare the energy recovered before and after the human-in-the-loop process), *NDCG* on the validation dataset, and Shapley values plots to analyze the patterns learnt by each model.

Semantic Grouping of Features and Evaluation

To facilitate the explanation and readability of this work, we exemplify the human-in-the-loop approach only on the visit-related features, including plots for the Shapley values in the training and test dataset. A brief description of these features is the following:

Types of Visits Most of the visit-related variables represent the visits made to the customers and their three possible results. More specifically, the *Fraud* features refer to the visits in which the company detected an NTL case. The *Correct* features refer to the visits where the installation was checked, but no NTL was detected. The *Impossible* features profile the visits with no conclusive result (in general, because the meter was not accessible). Finally, the *Visit* features represent all the visits without NTL/Non-NTL distinction. Based on this information, we profile the following features:

- *Number of Occurrences*: Those features that include the # prefix refers to the occurrences of that type of visit (e.g. #*Visit* refers to the number of visits the company has made to the customer).
- *Last occurrence*: The last occurrence of each type of visit is represented with the features with the *last* prefix (e.g. *LastVisit* would refer to how many months have passed since the last visit). When the customer has never been visited, the value of the feature is empty⁵.
- *Type of Visit*: A visit to a customer is often prompted by suspicion of fraud. In other cases, the visit is related to more generic reasons (e.g.

⁴ If the stakeholder decides to visit recidivist customer in July and August, and in September, we split the data considering the timestamp, in the test dataset we would have an over-representation of the recidivist customers.

⁵ That is, there is no value assigned, i.e. a missing value, that the Catboost library is able to process. This solution is applied in all the features to represent the non-existence of a value, e.g. the non-existence of a visit for that customer.

a generic revision of the meter). Both cases are reflected with suffix 1 and 2, respectively: *LastFraud1* refers to how many months have passed since the last fraud was detected in which the reason to visit was a suspicion of fraud (or NTL), *LastFraud2* corresponds to how many months have passed since the last fraud in which the reason to visit was not NTL-related. *LastFraud* (with no suffix) corresponds to the features that groups both types of features.

Region-Related Features There are also features related to the density of fraud *around* the customer. That is, *#FraudZone* indicates the historical number of NTL cases in a customer’s zone⁶. Similarly, *#FraudStreet* is the same information than the *#FraudZone* but focused specifically on the street where the customer lives, and *#FraudInBuilding* counts the historical NTL cases in the building where the customer lives. There exist for each feature a derivative (with a suffix *1Year*) in which the information is bounded in the last year (e.g. *#FraudZone1Year* indicates the number of fraud cases in the region during the last 12 months).

Threats There is a third group of features (*#threats* and *LastThreat*) that refers to the threats of the customer to the technician, i.e. if the customer violently prevents the installation revision from being carried out.

Energy Cut Finally, the *EnergyCut* feature indicates how many months have elapsed since the last energy cut by the company due to non-payment.

6.3.2 Tests

In this section we exemplify the process of stakeholder-system interaction by implementing the following: removing a feature due to its irrelevance, removing a correlated feature and correcting an outlier. We compare the baseline model and the resulting model in terms of *energy*₂₀₀ to see if, in addition to the improvement in terms of interpretability and bias reduction (that would help to increase the robustness in real campaigns), the resulting model also recovers more energy in the test dataset.

First Model (Baseline)

- *NDCG*: 0.44 in the validation dataset.
- *energy*₂₀₀: 249242.9kWh.
- Shapley Values: Figure 6.2 (training+validation model).

⁶ A zone corresponds to a technical term regarding the distribution of the electricity: nearby towns or neighborhoods in a big city share a zone.

Analysis As we can see in Figure 6.2, our baseline model has Shapley values that are abnormal, because the impact on the output is remarkably higher than all the other values for those features. For instance, if we analyze the *LastImpossible2* feature, there is no compelling reason to justify that a feature value increases the output of the model up to 25,000kWh, while the second highest Shapley value increases ten times less. Thus, this is an indicator of an outlier in the labelled information, i.e. an NTL case in which the company recovered a large amount of energy. In this case, the outlier corresponds to an NTL case in which the company recovered 260,000kWh, an extremely abnormal case of NTL due to the large amount of energy recovered⁷. With this information, the stakeholder would have two options: maintaining the outlier or correcting it. Maintaining an outlier could be useful in some specific cases (for instance, if the company aims to exploit biased patterns learnt⁸) but, in general, the stakeholder should consider its correction.

Next step In this case, an optimal solution would be reducing the weight of this NTL by modifying the label (for instance) four times (i.e. from 260,000 to 66,000kWh). With this change, we still indicate to the system that it is the higher NTL case in the labelled information, but we will avoid biases in the system.

Second Model (First Iteration)

- *NDCG*: 0.43 in the validation dataset.
- Shapley Values: Figure 6.3 (training+validation model).

Analysis First of all, we can see that we achieve a similar *NDCG* value in the validation dataset, i.e. it seems that the unbiasedness does not reduce the prediction capacity of our model. Then, the Shapley values from Figure 6.3 seem to indicate that the model learnt is better: there are no outliers (the higher Shapley value is reduced from around 30000 to 5000), and therefore it should generalize better on unseen data. So, in summary, a stakeholder would prefer this model over the previous one.

Next step For the next iteration we opt to drop the less important feature in the model: *#Threats*. This should not modify the model trained but would simplify the explanation provided to the stakeholders.

⁷ In the second NTL case in the dataset the company recovered around 50,000kWh. The typical customer consumption is close to 3,500kWh per year.

⁸ In some cases a biased pattern might be in line with business-related decisions. For instance, the stakeholder might consider not removing a pattern in which the customers from a region have higher predictions if the company aims to visit these type of customers more.

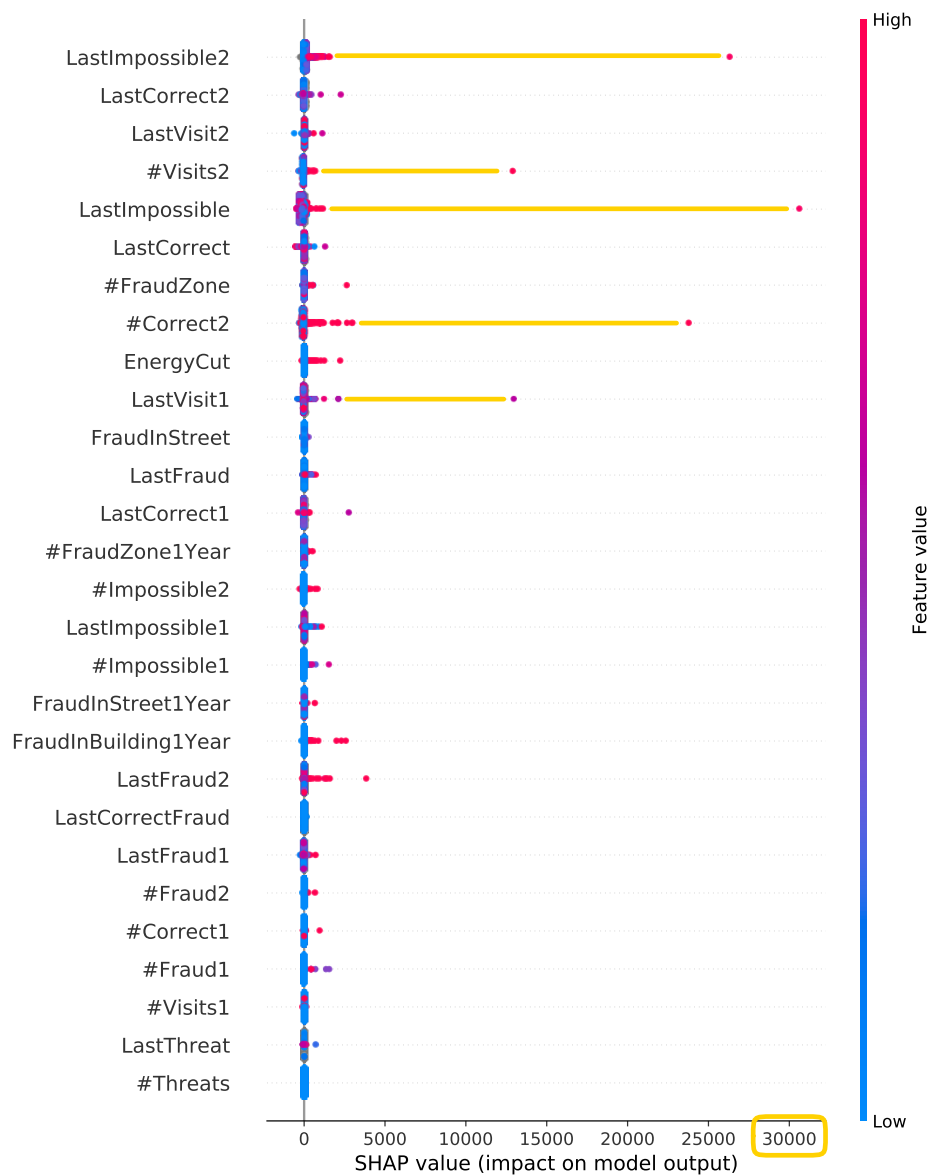


Figure 6.2: The outliers seen in the image (in yellow) are a consequence of an NTL case in which the amount of energy to recover is higher than 250000kWh. In this situation, the stakeholder in charge of the model building would consider reducing this prediction value to build a more unbiased model.

Third Model (Second Iteration)

- *NDCG*: 0.42 in the validation dataset.
- Shapley Values: Figures 6.4 (training+validation model) and 6.5 (top-scored customers from the test dataset).

Analysis Dropping the *#threats* has not changed much, as expected, what the model has learned (i.e. the plot from Figure 6.3 and the left plot from Figure 6.4 are similar). However, the possibility of dropping features can be fruitful from the company’s perspective. First, it allows to correct undesired patterns learned that, from the human perspective, have no logic but can be seen in a biased dataset. When we introduce a feature in the

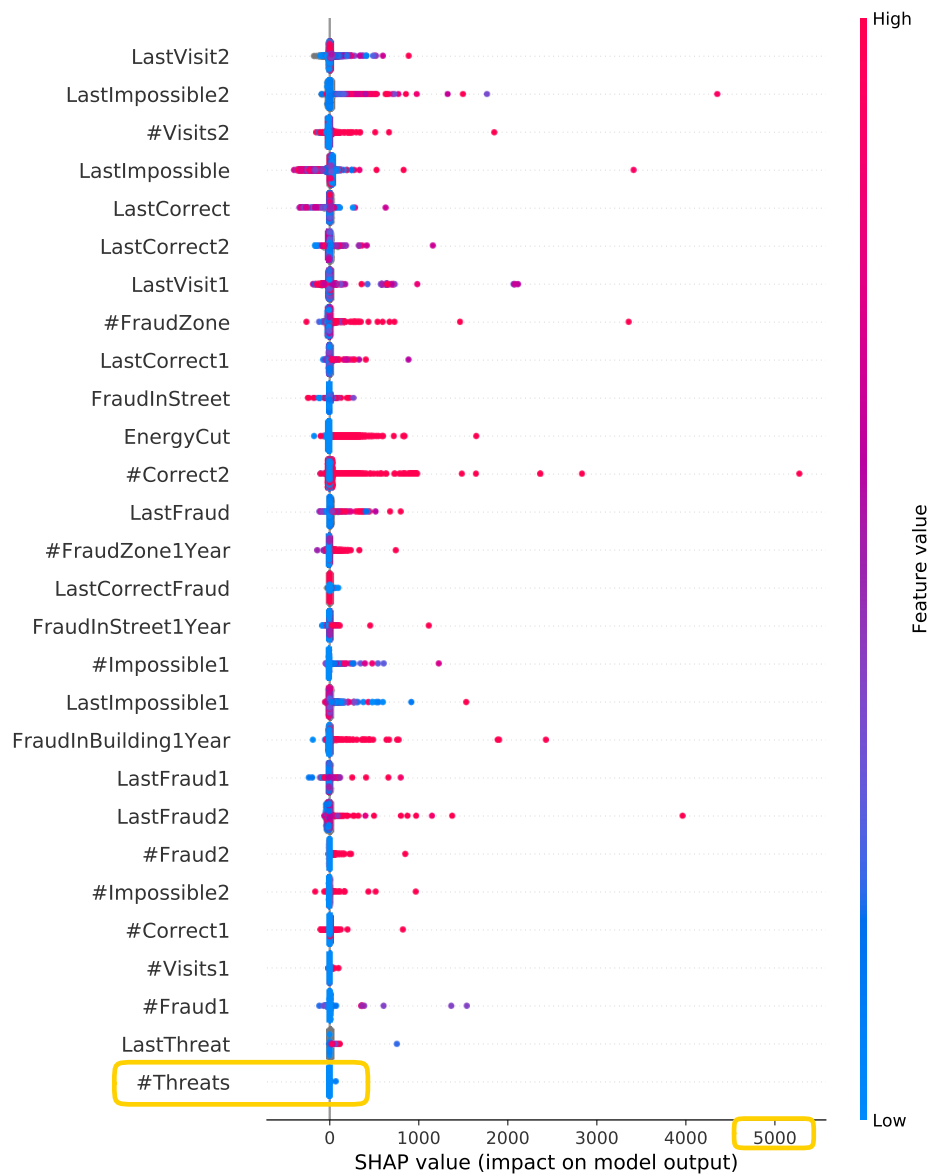


Figure 6.3: The Shapley values for the trained model indicates the non-relevance of the *#Threats* feature. Therefore, to facilitate the interpretation of the model by the stakeholders, we drop this feature from the training process.

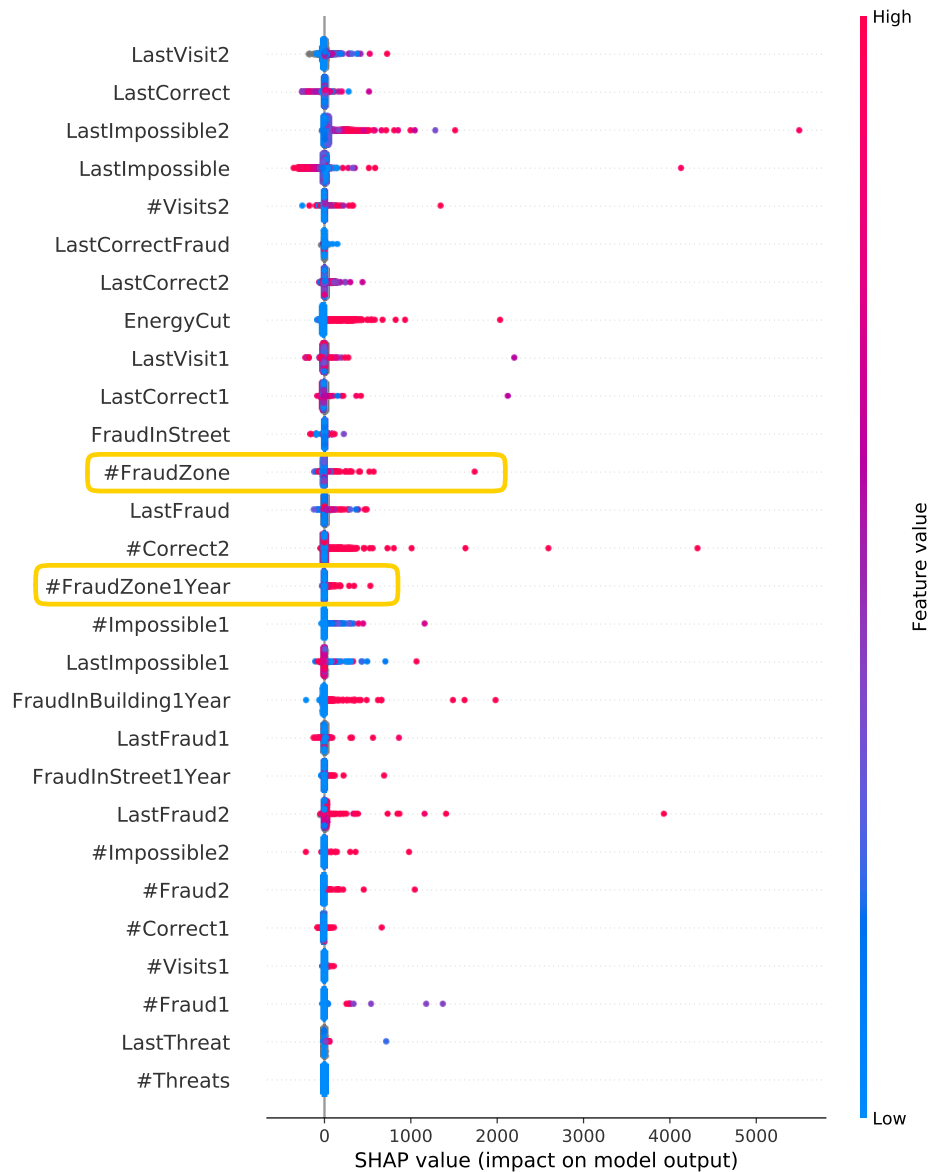


Figure 6.4: The Shapley values highlighted indicate that the model has learned similar patterns from both features with similar meanings. Removing one of the features would increase interpretability and reduce the curse of dimensionality. To decide the best feature to be removed, we can analyze how these patterns translate on unseen data (Figure 6.5).

system, we expect that the system will learn some specific patterns. For instance, when we profile with a feature that the customer is consuming much less than the average, we consider that the system should see this as an indicator of NTL. Therefore, if the system learns otherwise in one

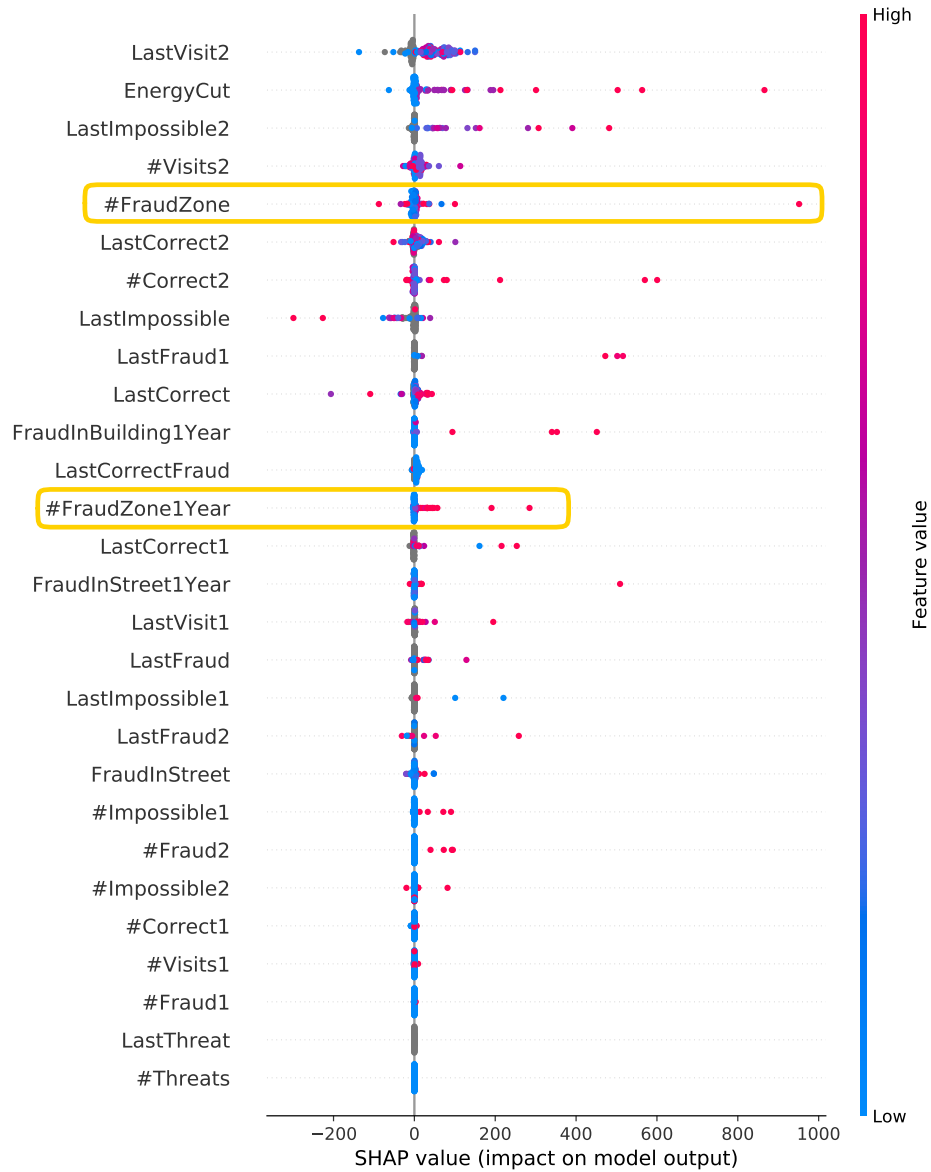


Figure 6.5: The Shapley values in the top-scored 200 customers test dataset we see that how *#FraudZone1Year* influenced the prediction is much clearer than in the *#FraudZone* feature and, therefore, we would drop the latter feature from the dataset.

specific domain⁹, the stakeholder can consider it appropriate to remove it in that specific campaign. Moreover, learning from fewer features with low relevancy helps avoid overfitting and increases the generalizability and interpretability of the model.

Next step For this third iteration, we exemplify the process of removing a correlated feature from the model. As shown in Figure 6.4, the features *#FraudZone* and *#FraudZone1Year* provide similar information to the learning process globally: a high number of NTL cases in the zone is an indicator of NTL. If we focus on the Shapley values from the top-scored 200 customers, we can see that the patterns learnt from the *#FraudZone* feature are unclear¹⁰ and, for this case, we would opt to remove the *#FraudZone* feature.

Resulting Model

The resulting model corresponds to the baseline model + correction of the bias + *#threats* drop due to its low relevance + *#FraudZone* drop (correlated with *#FraudZone1Year*).

- *NDCG*: 0.44
- *energy*₂₀₀: 257038.7kWh

Analysis The resulting model, in terms of *NDCG*, is as good as the vanilla model, and in terms of *energy*₂₀₀ is slightly better, recovering around 8000kWh more energy. However, in terms of Shapley values the resulting model is more trustworthy from the stakeholder’s point of view, and should generalize better on unseen data.

This example is rather naive since we have only slightly modified the system by implementing feature engineering. However, it exemplifies the benefits of the human-in-the-middle approach in which the stakeholder guides the system to learn an optimal model, mitigating the specific biases and other problems regarding the use of observational data. Moreover, the fact that the stakeholder is an active part of the system has positive consequences beyond the ones mentioned above (i.e. the better generalization on unseen data and the better interpretability), as the company can trust the system much more, one of the objectives of explainable AI [11].

⁹ If this undesired pattern is constantly learned in all the domains, then the feature drop would be definitive.

¹⁰ From the stakeholder’s point of view, it is simpler to explain the *#FraudZone1Year* pattern “high values is an indicator of NTL” than the patterns from *#FraudZone*, which are unclear, where sometimes a high value has positive Shapley values, and in other cases, it has negative Shapley values.

6.4 Discussion

One of the things we detected during the development of the system for Naturgy was the lack of "communication" between the system and the stakeholders: the stakeholders had a passive role in the system, they did not know what the algorithm was doing, nor could they participate in the learning process in an effective way. In this chapter we offer a human-in-the-loop solution that, despite its apparent simplicity, has great benefits for the system. On the one hand, the stakeholders can play a much more active role in guiding the learning system to make correct predictions and, on the other hand, its high transparency means that the company can have much more confidence in the implemented system.

Closure

In these chapters we have presented the process of developing a Non-Technical Losses (NTL) detection system for the company Naturgy. As in most of the literature, this system focuses on machine learning algorithms, more specifically on a supervised algorithm (Gradient Boosting Decision Tree), to learn NTL patterns from historical NTL/non-NTL cases to predict energy losses in the present.

The initial Chapter 3 provides the classical problem description seen in many other examples in the literature, i.e., a description of the data available, the creation of the classification problem, as well as some initial results that improved the existing rule-based approach of the company. However, we began to realize that the initial goal of achieving a robust, fully autonomous system that would achieve good results in any domain would be a major challenge.

From this initial chapter, we contribute to the NTL detection literature (and to the machine learning community applied to industry in general) by analyzing the use of explainability algorithms in an NTL detection system. During the development of this thesis, we have witnessed the birth of many explainability techniques, and the increasing interest in the artificial intelligence world for algorithms and methods that make predictions more transparent and fair. This work has been pioneering in implementing some of the most popular and efficient techniques to provide explainability to black-box algorithms, such as the Shapley values of SHAP or LIME. This highlights our work in the literature, since through explainability we propose solutions to existing problems in the use of observational data such as low energy recovery (Chapter 4) or how to involve the stakeholder in the process automated by the predictive algorithm (Chapter 6), allowing its guidance and the exploitation of human knowledge in an efficient way.

All and all, the development of this NTL detection system has provided us with an in-depth knowledge of explainable machine learning. In Part IV we discuss several aspects of the state-of-the-art techniques.

Part III

Explainable AI in Other Fields

Below is explained two collaborations in which the interpretability of the data science models have been a key aspect in their development.

Chapter 7 explains our collaboration with a project of the University of Padova and the Italian company My-Invenio. Our assignment consisted of providing explainability to an existing predictive process monitoring system, understanding why a prediction was being made and the important features of the system. This chapter is divided into two sections. The first section (Section 7) aims to provide an overview of Business Process Management (BPM), explaining the life-cycle of BPM, focusing on process monitoring. Section 7.2 analyzes the resulting paper from this collaboration ([58]). This work is a pioneer in successfully implementing explanatory algorithms in predictive process monitoring, automatically achieving explanations in line with those obtained by My-Invenio's human experts.

Chapter 8 analyzes the implementation of explainable solutions in machine learning for social science problems. Taking as example the initial results from the unpublished paper [95] from the Universitat de Barcelona in which we collaborated, we discuss if the explainable algorithms can provide enough information to validate hypothesis, replacing the regression models in social science literature.

Taking as an example the initial results of the unpublished paper [95] from the Universitat de Barcelona in which we collaborated, chapter 8 discusses the implementation of explainable solutions in machine learning for social science studies. We discuss whether explainable algorithms can provide sufficient information to validate hypotheses, replacing regression models in the social science literature.

Papers

Explainable Predictive Process Monitoring *International Conference on Process Mining* [58].

The Logic behind NGOs' Aid Allocation: a Complex Choice based on Past Decisions [95].

Chapter 7

Explainable Predictive Process Monitoring

7.1 Business Process Management (BPM)

7.1.1 Introduction

Business Process Management (BPM) is a discipline that encompasses different methods, tools and techniques that support and analyze how the processes in an organization work to guarantee their correct implementation and monitoring, and also their improvement when possible. Depending on the nature of the organization and its context, the improvement of the process might mean a cost reduction cost, a reduction in execution time or a reduction in error rates, but also the detection of innovation opportunities that would mean a competitive advantage against the competition [49].

The following are examples of the use of BPM in different fields:

Healthcare One of the paradigmatic examples of the benefits of the solutions that BPM can provide is its application in hospitals, where resources are limited, a rapid response to a medical emergency is mandatory, and the existence of different medical departments may lead to the existence of redundant processes due to a lack of interdepartmental communication. Thus, BPM aims to provide different solutions to improve hospital processes, including a better patient flow in terms of processing time, resource use and costs, but also optimization of less critical processes such as data gathering. Some examples of the analysis of using BPM in medical environments as well as their implementation can be read in [137, 129, 15].

Banking Another field in which there exist several examples of implementing Business Process Management solutions is Finance. For example, banks have had to modernize by leaps and bounds, moving in a short period of time from being entities where many of their processes were ex-

ecuted manually (and where the customer had to go to a bank branch to carry out any process) to doing these processes automatically or online. These changes involved analyzing the manual processes in order to modernize and automate them, minimizing the possible errors that could arise, reducing their time execution, and improving the overall customer experience. There exist several examples of successfully implementing BPM techniques in banking, e.g. [78, 122].

7.1.2 Life Cycle

Business Process Management comprises several phases that guide the process of understanding the existing processes in organizations, their flaws and problems, their improvements and post-control; this life cycle of a process is usually represented as seen in Figure 7.1, where each state is defined as follows [49]:

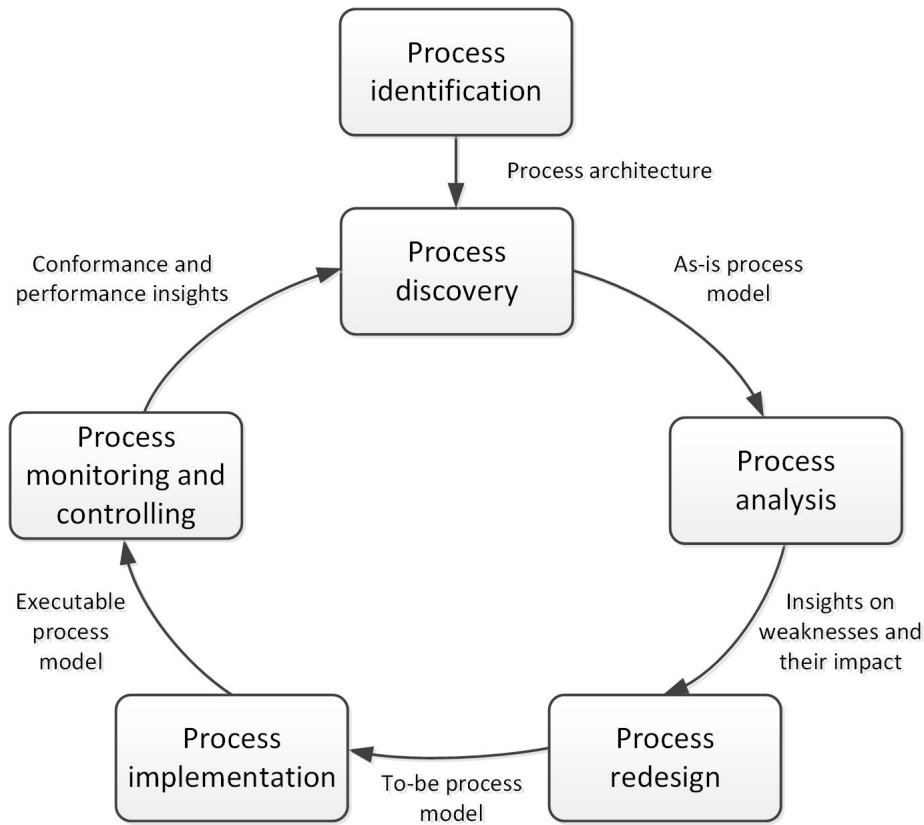


Figure 7.1: Lifecycle of Business Process Management.

Process Identification Process Identification corresponds to the initial process of BPM when the organization has not engaged in BPM before. The result of this stage is a process architecture that provides an overall

vision of the company's processes and their relationship; identifies which processes should be analyzed and improved. To achieve this process identification, it is necessary to determine how the process should be evaluated, i.e. the process performance measures (or metrics), e.g. the duration of a process, the quality of the output or the errors committed. Establishing the optimal measure is crucial for BPM to analyze the correctness of the process properly.

Process Discovery The next step corresponds to properly understanding the business process in detail, where the outcome of this process is the process models. These process models must be understandable since they are a key aspect of all the stakeholders involved in the BPM analysis. The model can be text descriptions, a diagram, or a combination of both. The diagrams are usually flowcharts where the rectangles represent activities, and diamonds represent points in the process where a decision has to be made. From this baseline, several extensions of flowcharts exist that include different aspects that help to represent the process better.

Process Analysis Once the process is appropriately represented and explained, the next step consists of analyzing it to identify the issues in the process. In this step of the process, the correct determination of the performance metric would help detect the current method's errors and problems. The issues identified are then classified according to their importance (i.e. what would be the impact of solving them, or the effort needed to do it).

Process Redesign The process redesign corrects the problems detected in the previous step. This step requires a deep understanding of the global process, since in many cases a solution to the problems of the current process might cause other problems that currently do not exist. In point of fact, the process redesign is closely related to the previous step to avoid these problems. The new process design should improve the performance metrics established at the beginning of the process.

Process Implementation After the process redesign has been done, it should be implemented. This involves changes from the workers, referred to as organizational change management, in their activities and methods used, and technical changes referred to as process automation, such as a new IT system.

Process Monitoring Once the process is redesigned and implemented, the process should be monitored to record its performance. This information should be then used in the short term to implement small corrections to the model built, and in the long term, to restart the cycle. These event logs can be used to monitor the historical performance of the process executions (referred to as Offline Process Monitoring) and monitor processes

currently being processed (referred to as Online Process Monitoring). The former analyzes information from a long period of time (e.g. a full year) to provide a picture of the performance of the process, and is useful to understand the correctness of the process globally. The latter is focused on analyzing current process instances and is used to correct specific problems (e.g. when a customer request remains unanswered).

7.1.3 Predictive Process Monitoring

Predictive Process Monitoring is a sub-field of the Process Monitoring process in which the event-logs from the process are used to generate predictions for a specific process, for instance the remaining time of that process (that would correspond to a numeric output), the fulfilment of a certain goal (binary), or a metric that would score the effectiveness of the process.

The initial stage of Predictive Process Monitoring consists of encoding the process' event logs into feature vectors. In general, the information available is massive and, therefore, the encoding process requires feature engineering to extract the most relevant information from four perspectives [43]: control-flow perspective (i.e. the order of the activities performed in the process), the data-flow perspective (i.e. the different attributes attached to the event), the time perspective (i.e. the duration of an activity or the whole process) and the resource perspective (i.e. information regarding the organization that executes the process). These four perspectives differentiate the data processing for process monitoring from the more generic data models (e.g. supervised machine learning).

The next stage corresponds to the building of the predictive model. Depending on the approach, the model can be a statistical model (e.g. Hidden Markov Model, [79]), a machine learning approach (e.g. [93]), an annotated transition system (e.g. [120]) or a hybrid of different approaches (e.g. [135]). In the same way, the models can be classified as process-aware models (if there is an explicit representation of the process model, e.g. the annotated transition models) or non-process-aware models (if there is no explicit representation of the process model, i.e. mostly the machine learning models). The validation of the model, similar to what is done in classical machine learning, is evaluated using classical metrics such as precision or accuracy for classification problems, and RMSE or MAE for regression. Finally, the model is used for predicting. In general, predictive process monitoring uses the models for online predictive monitoring (e.g. to predict the remaining time of a process) at a certain point of the process, referred to as checkpoint.

Most of these prediction systems cover the following purposes:

- Performance indicator: The Process Performance Indicators are metrics that evaluate how well (i.e. how efficient and effective) the process is, providing quality information useful in controlling and improving it. The most common performance metrics aspects evaluated

are time-related (e.g. the duration of the activity).

- **Risk Predictions:** Risk predictions encompass all information related to the detection of anomalous process executions, either because an instance of a process is taking too long to execute, or because an activity is running too many times (in a loop).
- **SLA violation predictions:** A Service Level Agreement (SLA) defines a contract between a supplier and a customer, where a breach of any part of the contract may result in financial penalties. Thus, predictive models can be used to identify if an SLA is going to be breached in order to correct it in time.

In addition to these cases, there exist other purposes such as the detection of abnormalities (e.g. the abnormal termination of an event) that do not fit previous categories but are common in predictive process monitoring.

7.1.4 Predictive Process Monitoring through Machine Learning

Recently, the use of machine learning techniques in predictive monitoring has been increasing dramatically. Their application is a clear example of how machine learning can be implemented in the industry.

There are many examples in the literature, ranging from cases where unsupervised and supervised algorithms are used. Some examples of the application of unsupervised algorithms are [44] (that describe a clustering and decision method for the prediction of a violation (or otherwise, the fulfillment) of a determined predicate, or [54], which implements a solution based on clustering to determine a violation of the ongoing instance. In relation to the examples of supervised algorithms, we find examples of both interpretable algorithms (such as [133], in which the framework presented combines different techniques, including Decision Tree models, to predict risk probability), and black-box algorithms that use more complex algorithms that, a priori, should better exploit large amounts of data. Examples of the latter would be the predictive process monitoring systems that are based on Long-Short Term Memory, e.g. [132, 93].

As we can see, machine learning offers different solutions for predictive process monitoring, so the choice of algorithm will depend on many factors. As we are analyzing in this thesis, one of the factors to consider is that the use of complex algorithms is, according to the accepted literature, the algorithms that can predict best but are poorly interpretable. This is a problem as we have discussed above in the case of NTL detection in electricity. In the case of BPM the fact of using opaque algorithms is also a problem to be taken into account, since for sure many stakeholders with different roles (that need to understand the predictive systems) will be involved in the process of improving the institution's processes.

7.2 Case Study: Explainable Predictive Process Monitoring

In [58] the problem of providing trustworthy explanations to a predictive process monitoring framework is tackled. This work builds a framework based on LSTM models that estimates the corresponding KPI value with an explanation of the features that influenced that prediction.

7.2.1 Preliminaries, Context and Problem Statement

The Long-Short Term Memory algorithm is the approach that, according to [97, 93, 132], provides better results, at the cost of not being transparent. At the time of publication of the paper no relevant work had managed achieved to provide explainability of predictive monitoring models, as explained in [84], which claims that *“little attention has been given to [. . .] explaining the prediction values to the users so that they can determine the best way to act upon”*, and that *“it is necessary to develop tools that help users to query these models in order to get information that is relevant for them”*. Existing work attempted to provide explanatory solutions [21, 109], but they were simple solutions bounded to simple and specific use cases, difficult to generalize in real case scenarios.

In [58] the proposal is to implement explainability with Shapley values to an LSTM-based predictive monitoring framework. The main dataset¹ used in the work corresponds to an Italian banking institution, and the process to be monitored is the *Bank Account Closure* (the process that deals with the closure of customer’s accounts), with 212721 events and 32439 complete traces. From this dataset, 2/3 of the traces would be the training dataset (20% of which has been used for hyperparameter optimization, i.e. used as validation dataset), and 1/3 corresponds to the test dataset.

The KPI to be predicted related to this process are the following:

Remaining Time The estimation of the remaining time allows the bank to detect those process cases requiring special attention in order not to postpone them much longer.

Activity Occurrence There exist several activities² that are linked to contingency actions that should be avoided to reduce inefficiencies in terms of cost, time and resource utilization.

Case Cost In this case, the estimation of the case cost would help to detect cases that require special attention.

¹ The original paper contains other experiments with public datasets.

² *Authorization Requested, Pending Request for Acquittance of heirs, and Back-Office Adjustment Requested.*

The LSTM model provides an optimal structure to represent the sequence of events that constitute a trace. Being e an event (encoded as a feature vector $e = \langle x_1, x_2, \dots, x_n \rangle$), then a trace t can be represented as an ordered sequence of events (i.e. $t = \langle e_1, e_2, \dots, e_m \rangle$), that is a matrix $N \times M$ of features. Therefore, the Shapley values provide, for each feature, a value indicating its contribution to the predictive output in the same $N \times M$ form. Regarding all the Shapley values, in this work the Shapley values considered relevant are the values outside the range $I = [\mu - \delta\xi, \mu + \delta\xi]$, μ being the average of the Shapley values, ξ their standard deviation and δ a parameter set by the user. Once the relevant features are computed, two types of explanations are provided, the offline and the online explanations, summarized as follows:

Offline Explanations Heatmap that overviews information regarding the frequency in which an explanation has been relevant for the test dataset. Therefore, this offline explanation provides a global explanation of the importance of each feature.

Online Explanations The online provides information regarding running instances. In this case, the information is given as a table.

7.2.2 Case Study Analysis

Below we provide a summary of the experiments from [58], results that show not only the accuracy of the LSTM models but also the good explanations provided by SHAP.

Remaining Time

According to the heatmap from Figure 7.2, the most influential feature would be *CLOSURE_TYPE!=Inheritance*, with a negative value of -71598, i.e. it normally reduces the remaining time. This pattern can be humanly validated as when the type of procedure is *Inheritance*, the process duration is 29 days, versus 14 days when not. Similarly, the LSTM model detects that attributes *Role=Back-office* and *CE_UO=BOF* (back-office activities) reduce the prediction time, an expected behavior since both activities are performed in the final stage of cases.

Regarding the online explanations, in Table 7.2 there is an example of the information provided by the framework. For instance, the last row indicates that the remaining time of that case is just over two and a half hours, with two explanations increasing the prediction, one related to the fact that the previous activity performed was not *Service Closure Request with BO Responsibility*, and the other related to the resource performing the previous activity with a role not being *Back-Office*.

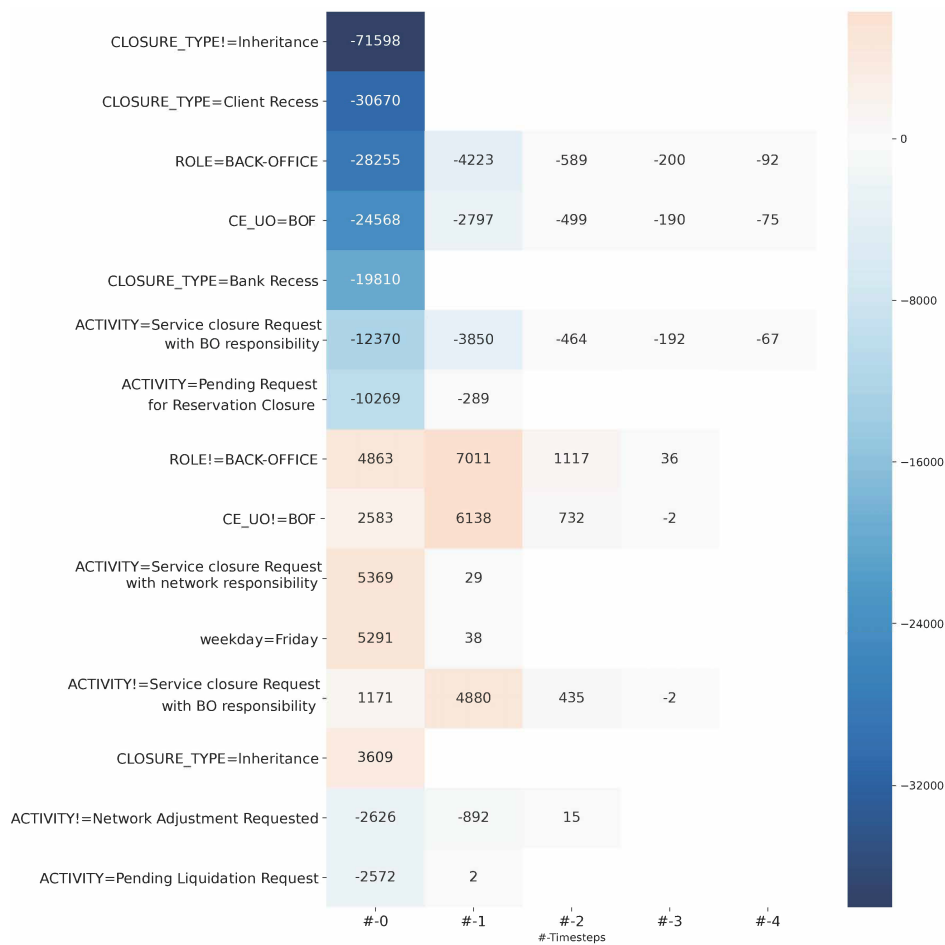


Figure 7.2: The offline explanation of the remaining time KPI.

CASE ID	REMAINING TIME	EXPLANATIONS FOR INCREASING REMAINING TIME	EXPLANATIONS FOR DECREASING TIME
201810011258	5d 6h 7m	ACTIVITY=Evaluating Request (NO registered letter)	CLOSURE_TYPE!=Inheritance
201810000206	5d 2h 12m	ROLE=DIRECTOR	CLOSURE_TYPE=Bank Recess
201811010829	2d 2h 31m	ROLE!=BACK-OFFICE (-1) AND ACTIVITY!=Service closure Request with BO responsibility (-1)	-
...

Table 7.1: Online explanations for *Remaining Time* for three running cases. When the explanation is followed by (-1), it means that it refers to the value assigned to the attribute by the event that precedes the last of respective case.

Table 7.2: Online explanations for *Back-Office Adjustment Requested*. Values 1 and 0 indicate if the activity is predicted to occur or not. Explanation followed by (-1): attribute value assigned by the event that precedes the last of respective case.

CASE ID	BACK-OFFICE ADJUSTMENT REQUESTED	EXPLANATIONS FOR BACK-OFFICE ADJUSTMENT REQUESTED HAPPENING	EXPLANATIONS FOR BACK-OFFICE ADJUSTMENT REQUESTED NOT HAPPENING
201810000206	0		ACTIVITY=Service closure Request with network responsibility (-2) AND CE_UO=195 (-1)
201811008237	1	CLOSURE_TYPE=Porting	-
201812005701	1	CLOSURE_REASON!=1 - Client lost	-
...

Finally, the metric used for the LSTM was the *Mean Absolute Error* (MAE), achieving a MAE on the test dataset of around 4.37 days, which is around 28% of the average case duration (i.e. 15.5 days).

Prediction on Activity Occurrence

In this work the contingency action analyzed is *Back-Office Adjust Requested*, an activity related to inefficiency that demands rework. The metrics regarding the LSTM model that predicts this activity achieved an F1-score of 0.65, an AUROC of 0.86, and APR of 0.69.

In Figure 7.3 we have the heatmap from the offline analysis, where the Shapley values indicate that the features related to a closure of the bank account influence the most. When the customer makes a request to close all their bank account (i.e. *Closure Reason=1 - Client Lost*) or only one bank account (i.e. *Closure Reason=2 - Keep bank account- Same dip*), the adjustment is unlikely: the values of these activities in the heatmap are negative. Negative values are seen in other similar features where the customer decides to close their bank account. On the other hand, when the bank is the one that decide to close the account (e.g. *CLOSURE_TYPE=Bank Recess*), then the activity is more likely to occur.

In Table 7.2 we have a small sample of the online explanations. Taking the first case as an example, we can see the rework activity is not expected to happen because two events ago *Service Closure Request with Network Responsibility* has been performed and because the previous event has been performed by the resource 195. On the contrary, it is predicted to eventually happen for the other two cases in the table, and the explanation is related to the closure type being Porting and the closure reason not being Lost Client.

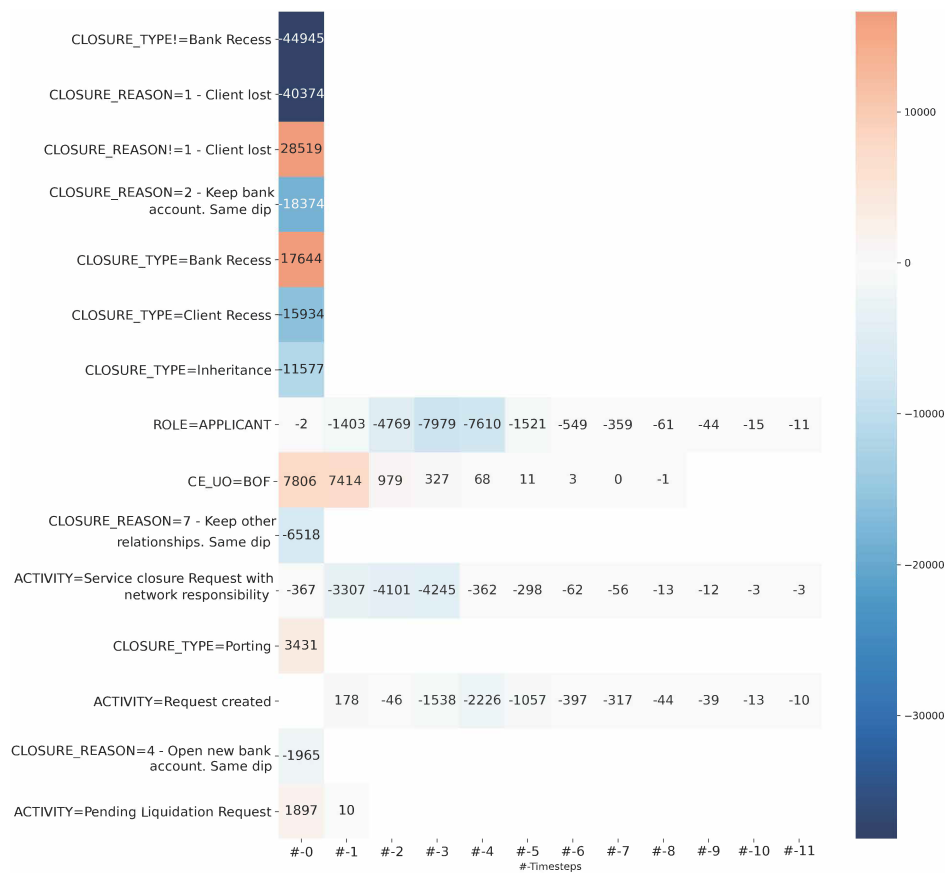


Figure 7.3: Offline explanations for *Back-Office Adjustment Requested*

Case Cost Prediction

Finally, for this third KPI the metric adopted is the MAE, achieving an error of 0.95 euros. In this case, this error is very low, since the average cost is 12.86 euros, with a standard deviation of 6.41 euros.

Figure 7.4 shows the heatmap of the offline explanation. In this case, the fact that a bank account needs to be closed as requested by the customer (i.e. *CLOSURE_REASON=1 - Client Lost*) increases the cost of the process: according to the paper this is a consequence of the fact that this process is evaluated by the director whose wage is higher than other workers. On the other hand the values in the heatmap are negative when the bank is the one that closes the account (*CLOSURE_REASON=1 - Client Lost*) or the customer only wants to close one of its bank accounts (*CLOSURE_REASON=2 - Keep bank account. Same dip*), expected value considering that the director does not have to carry out any evaluation in these cases.

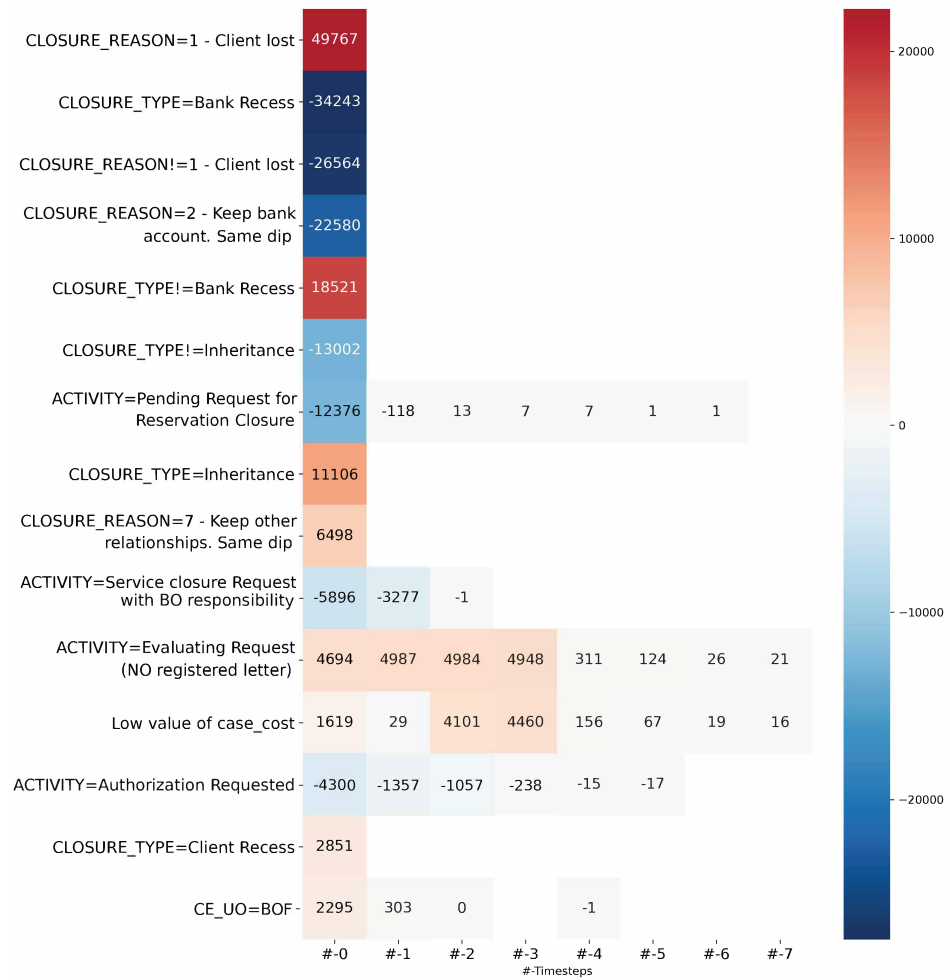


Figure 7.4: Offline explanations for *Case cost*

Closure

In predictive monitoring, a lot of work has been done to increase the accuracy of the predictive monitoring system, usually by implementing more complex solutions, such as using black-box algorithms. However, little attention has been paid to making the solutions transparent, facilitating the interpretation of the predictions by the stakeholders, who have to make decisions based on these values. In a way, we find ourselves in a situation similar to the one explained in the project implemented for Naturgy, where automation through artificial intelligence can generate distrust among stakeholders due to its opacity. The lack of interpretability can lead one to discard the use of more complex solutions that, in general, should provide better results, in favor of simpler solutions. Therefore, it is necessary to build trust to guarantee that the system is adopted in practice [94, 46].

In this chapter we have analysed the work done in [58], which implements an LSTM framework to predict different KPI. This paper achieves a milestone in Business Process Management in general, and in predictive monitoring in particular, by implementing a fully interpretable system that combines the algorithm that, according to the literature, is the best option to guarantee high accuracy, with the Shapley values that provides reliable explanations, in line with the on obtained from the human analyst from MyInvenio.

Our contribution focused on the development of Shapley values explanations of the predictive framework. In Part IV we provide different conclusions regarding the use of this solution in a Business Process Management context, and in Machine Learning in general.

Chapter 8

Explainable Black-Box Algorithms in Social Science

8.1 Data in Social Science

8.1.1 Data Modelling in Social Science

Social Science is the field in science that studies societies and their individuals, including political structures (i.e., political science), the value of goods and services (i.e., economics), or past human events (i.e., History), among many other research fields.

In general, social scientists aim to validate hypotheses through data. That is, the social scientist builds a statistical model with some independent variable (i.e., explanatory variables, that represent different hypotheses) and a dependent variable (i.e., the response). The model reveals relationships between the independent variables and the dependent variable. Based on these variable interactions, the social scientist can validate one hypothesis or another. The quintessential social science model is linear regression (and other generalized versions, e.g., logistic regression) used in Gaussian data and linear, where the coefficients indicate how an independent variable influences the dependent variable. Based on these coefficients, the hypothesis can be validated or otherwise. Thus, the main focus in social science is to validate a hypothesis through statistical analysis.

In contrast, predictive machine learning aims to build a model that automatizes a prediction process by learning historically labelled instances to predict unseen new instances. The data distribution is unknown, requiring that both the labelled and unseen instances to be drawn with the same distribution. The fact that in many cases there is no prior assumption of the data distribution means there is no "correct" modelling algorithm, and therefore it is necessary to analyze how well a model predicts using training-validation split or cross-validation, as explained in [2.1.2](#). The best model in terms of benchmarking (e.g., the precision result in a validation dataset) is considered the best option to automatize the prediction pro-

cess. This is another key difference with the social science models where benchmarking (e.g., the R^2 and the p value in linear regression, Section 2.2.1) is used to validate that the model is able to explain the variance of the output, but there is usually no direct comparison between approaches.

The intrinsic characteristics of Machine Learning made the field exploit the large amount of data available thanks to Big Data, since more data should better represent the reality that one is aiming to model. However, this is not seen in Social Science, where the humanistic point of view of exploiting the data, where the main goal is to detect causal patterns to validate hypotheses, has kept them in the small data.

8.1.2 Challenges and Possibilities of XAI in Social Science

The use of artificial intelligence solutions outside the purely technological field has been evident for years. For example, we have already mentioned in Chapter 1 its use in the medical field, being one of the pioneering fields in applied artificial intelligence. In social sciences, its use is less evident, although there are several examples of it. In [113] how artificial intelligence has been introduced in social and behavioral science between 2010 and 2019 is analyzed, differentiating three different applications: to increase the effectiveness of diagnosis and prediction of different conditions (e.g. by reducing the diagnosis of autism [140], the risk of alcohol use among adolescents [2]), to increase understanding of human development and functioning (e.g. Twitter analysis to examine weekly trends in work-related stress and emotions [141]), or to increase the effectiveness of data management in different social and human services (e.g. detecting child abuse cases [8]). All these applications are focused on implementing applied solutions to *social* problems. However, extracting conclusions and validating hypotheses from complex machine learning is not intuitive.

Recently, as we explained in Section 2.3, the artificial intelligence community is putting a lot of effort in implementing solutions to better understand predictive algorithms and, therefore, mitigate many of the existing ethical and technical challenges. Hence, these explanatory solutions (that in this thesis have already been introduced in previous technological projects) may be the necessary tool to introduce the use of complex artificial intelligence algorithms to analyze social science problems from a theoretical perspective; these explanatory algorithms would allow us to validate or discard a hypothesis, as done when using interpretable algorithms, but with the added value that complex models can represent reality in a more complex way, thus bringing new insights into analyses.

8.2 Case Study: How NGOs Prioritize Foreign Aid Recipient Countries

The work done in [95] analyses the reasons why an NGO goes ahead with a project in a country, proposing the hypothesis that their previous experience (i.e. if the NGO has a bond with the country) is the main reason in their decision, a hypothesis not explored in the literature. This work analyse this hypothesis with other existing hypotheses in the literature, using as modelling algorithm the classical logistic regression approach, but also explores the use of explained machine learning to answer this question.

8.2.1 Preliminaries, Context and Problem Statement

In general, the literature proposes different hypotheses when determining the reason why an NGO develops a project in a country, which can be grouped into two main hypotheses: the pragmatic rational and/or normative and reasons of principle [22]. For the former hypothesis, most of the literature considers that the NGOs usually allocate their projects based on donor prioritization. The latter considers that the NGOs are principle-driven organizations and, therefore, their aim is to reduce poverty and inequality.

The work done in [95] puts on the table a third hypothesis: that the NGOs are path-dependent, developing projects where they have bonds with the community or other local NGOs, since they have the knowledge and experience to develop their projects with success. This work explores the three hypotheses in the Spanish context. The data from this research is extracted from different data sources, as seen in Table 8.1, profiling different Spanish NGOs (see Table 8.2). Each data instance profiles the NGO and the country from 2009 to 2016, as well as the existence (or the lack thereof) and information of a project developed by the NGO in that country for one specific year. Except for the control variable *colony*, the variables aim to represent different aspects of the hypothesis explained above. That is:

H1: Back Donor Effect This hypothesis, defined in the work as *NGOs' aid allocation is influenced by public donors' preference. In other words, NGOs prioritize aid to those countries that prioritize their own donor* is represented in the profile with both the *Donor Aid Budget* and *Public Grant* variables.

H2: Country Needs The second main hypothesis, which argues that *Objective country needs, such as poverty indicators, increases the probability that NGOs provide their aid to that country*, represented in the work with the variable *GDP per capita*, has two derived sub-hypotheses, i.e. *The recommendations of international organizations to pay more attention to special countries will positively influence NGO countries prioritization.*,

profiled by the variable *UN LDCs*, and *NGOs' aid allocation is influenced by their geographical preferences as stated in their statutes*, represented by the *Latin America Mission* and *Africa Mission* variables.

H3: Path dependence the new hypothesis defended in the thesis corresponds to two ideas, i.e. *Allocating aid repeatedly in one country increases the probability of coming back, i.e., accumulating experience in one country predicts an NGOs aid allocation*, profiled by the *Budget Previous Year* variable, and *Choosing one country as aid recipient at present is highly influenced by the factor that the NGO has put down roots in that country, i.e., has an institutional structure in the country such as a delegation*, represented by the *Delegation* variable.

Description of the independent variables

VARIABLE	DESCRIPTION	DATA SOURCE
Donor Aid Budget	The total amount of euros allocated by Spain to projects, agreements and actions for each of the priority countries and year.	Agencia Española de Cooperación Internacional para el Desarrollo (AECID): https://www.aecid.es/ES/la-aecid/nuestros-socios/ongd/ (accessed: May 2018)
Public Grant	Amount of euros granted in public subsidy to NGOs in competitive and annual calls. Subject to geographical prioritization criteria set by the State.	Agencia Española de Cooperación Internacional para el Desarrollo (AECID): https://www.aecid.es/ES/la-aecid/nuestros-socios/ongd/ (accessed: May 2018)
GDP <i>per capita</i>	GDP per capita (constant 2010 \$) of each country. Information extracted from the World Bank Database. The value is imputed when this information is not available.	World Bank : https://data.worldbank.org/indicator/NY.GDP.PCAP.CD , Datos Macro Expansion (for imputation values): https://datosmacro.expansion.com/
UN LDCs	Binary variable indicating if the country is in the UN's list of Least Developed Countries.	UNCTAD, 2012 https://unctad.org/system/files/official-document/ldc2012_en.pdf (Accessed, May 2018)
African Mission	Dummy variable that indicates if the NGO prioritizes African Countries.	NGO founding statutes
Latin America Mission	Dummy variable that indicates if the NGO prioritizes Latin American Countries.	NGO founding statutes
Budget Previous Years	The total amount of euros spent by the NGO on projects in a particular country in the previous year.	Coordinadora de ONG para el Desarrollo-España (CONGDE) https://coordinadoraongd.org/
NGO Delegation	Binary variable indicating if the NGO has a Delegation in the country.	Coordinadora de ONG para el Desarrollo-España (CONGDE) https://coordinadoraongd.org/
Colonial history	Control variable that indicates if the country is a former Spanish colony.	Wikipedia

Table 8.1: Description of the variables and data source

Name of the NGOs included in this work and their size

NGO NAME	SIZE
Acción Contra el Hambre	Large
Acción Verapaz	Large
Caritas Española	Large
Cruz Roja	Large
Médicos del Mundo	Large
Oxfam Intermon	Large
Asociacion FONTILLES	Medium-large
Educo	Medium-large
Fundación Adsis	Medium-large
Fundacion Entreculturas	Medium-large
Manos Unidas	Medium-large
Movimiento por la Paz -MPDL-	Medium-large
ADRA	Medium
AIETI	Medium
ALBORAN	Medium
Amigos de la Tierra España	Medium
Asociación Entrepueblos	Medium
CESAL	Medium
CODESPA	Medium
Economistas sin Fronteras de España	Medium
FAD	Medium
Farmamundi	Medium
FERE-CECA	Medium
Fundación CIDEAL	Medium
Fundacion de Religiosos para la Salud	Medium
Fundacion del Valle	Medium
Fundacion Iberoamerica-Europa	Medium
Fundación para el Desarrollo de la Enfermería	Medium
Fundacion Promocion Social	Medium
InteRed	Medium
ISCOD	Medium
JOVENES Y DESARROLLO	Medium
Juan Ciudad ONGD	Medium
MUNDUBAT	Medium
Paz con Dignidad	Medium
Prosalus	Medium
PROYDE	Medium
SED	Medium
Accion Verapaz	Small
AMREF Salud Africa	Small
Edificando Comunidad de Nazaret	Small
Farmacéuticos Sin Fronteras de España	Small
FISC-COMPAÑIA DE MARIA	Small
PUEBLOS HERMANOS	Small

Table 8.2: NGOs included in the study classified according to the number of employees (E) or annual income in million Euros (AI). **Small:** $< 10E$ or $AI \leq 2$. **Medium:** $< 50E$ or $AI \leq 10$. **Medium-large:** $< 250E$ or $AI \leq 50$. **Large:** $\geq 250E$ or $AI > 50$.

The resulting dataset consists of 5236 developed projects. The negative cases (i.e., the absence of a project from an NGO in a year in a country) are built combining all the possible NGO-country combinations that have no project. Finally, each NGO has 139 instances each year, with a total information of 1112 instances per NGO. The resulting dataset includes 44804

negative instances (i.e., triads $NGO, Year, Country$) with no project, for a total of 50040 instances. With this dataset the work proposes two models to analyze the hypothesis: a logistic regression and an LSTM model (combined with Shapley values to understand the patterns learned).

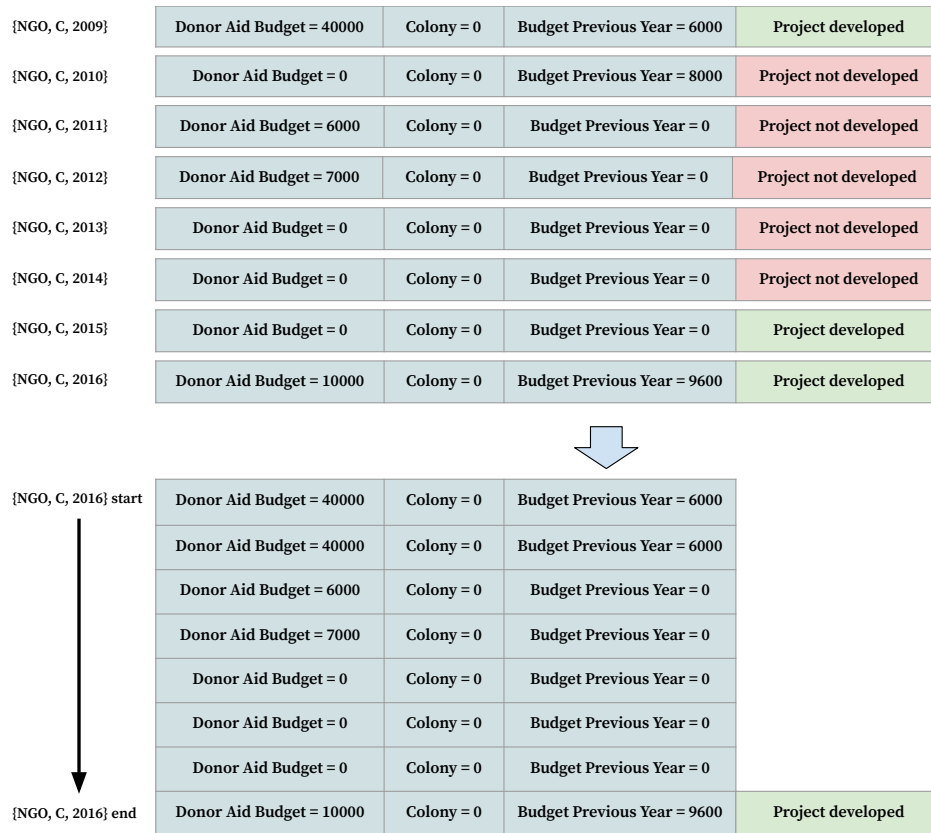


Figure 8.1: Our regression dataset considers each $NGO, Country, Year$ instance as independent, with no relationship between them. For this reason, each instance has a dependent variable (i.e., if the project was developed or otherwise). In contrast, the LSTM dataset has a temporal structure in which only the $NGO, Country, 2016$ has a dependent variable, and all the information from previous years (from 2009 to 2015) is information from the instance $NGO, Country, 2016$.

The regression model is chosen because it is widely used in the literature (e.g., all the references included in this section use regression models to validate their hypothesis). However, the paper goes a step further beyond what exists in the literature and analyzes the data using the LSTM model to have the data structured as a time series to further analyze hypothesis 3. This analysis forces a different representation of each instance. More specifically, for each NGO and country we will have as dependent variable if the NGO developed a country in 2016, where the independent variables will be all the historical information from that country and NGO , as explained

in Figure 8.1.

8.2.2 Case Study Analysis

Here below we provide a summary of the analysis included in [95].

Previous Analysis

Before data modelling, a previous analysis is required to not only understand the characteristics of the data but also to avoid collinearity. In Figure 8.2 the Spearman correlation between variables is analyzed. The Spearman correlation is chosen over the Pearson Correlation due to the characteristics of the data (the assumptions of normal distribution, linear relationship and no homoscedasticity are not met in our data), but also to achieve a comparable correlation metric between two continuous variables, a continuous and a binary variable, and two binary variables¹.

Three pairwise correlations between the independent variables can be highlighted in Figure 8.2: the negative correlation of -0.65 between the UN LDCs and the GDP per capita variables (that indicates that the UN prioritizes the poorer countries); the correlation between the Delegation and the Budget Previous Year of 0.5, indicating that there is a statistical dependence between the existence of a project in the previous year and the existence of a Delegation; and the Donor Aid Budget and the Colony variable correlation (0.47), indicating that Spain seems to prioritize historical colonies (e.g. Latin American countries) in their donor aid. No correlation seems important enough to implement feature selection due to multicollinearity, as confirmed with Table 8.3 with very low Variance Inflation Factor values between 1 and 2.

When analyzing the correlation between each independent variable and the dependent variable we see that the H3 variables have an above-average correlation with the dependent variable. However, there exist no leakage in terms of the chronological order², and also, the variables are not good enough to fully explain the prediction, as explained in Table 8.4. Therefore, both features are simply considered as good predictors and kept in the model.

Logistic Regression

Table 8.5 summarizes the logistic regression results, including one model for each hypothesis, and a fourth model that analyses the importance of

¹The Spearman correlation measures the monotonic relationship between continuous data and/or ordinal data based on the ranked values instead of their raw value. If it is assumed that the binary data can be seen as ordinal data of two levels, then it is possible to obtain a global vision of the correlations between the variables.

²

Spearman's Correlation Heatmaps

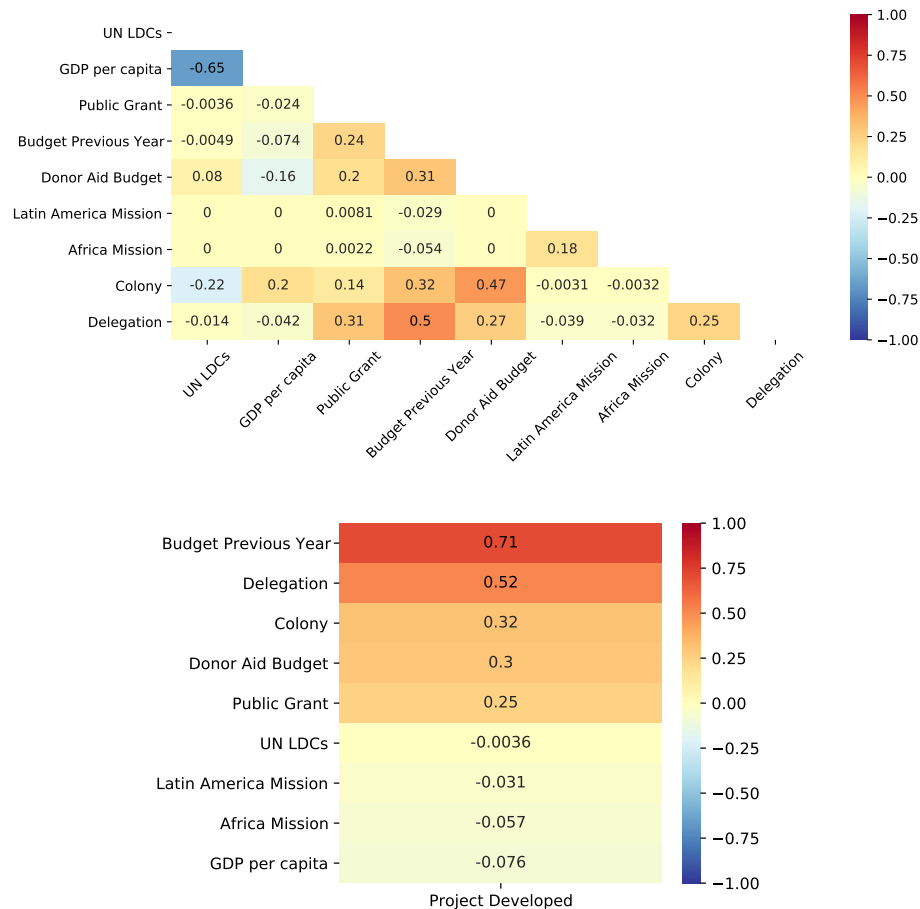


Figure 8.2: Spearman's Correlation between variables. The plot above provides the strength and direction of monotonic association between the independent variables. Most of the correlation values are low, with some correlation values around ± 0.5 . The plot below provides the same information between the independent and the dependent variables, showing that the variables from H3 hypothesis are more correlated to the Project Developed variable than the variables from H1 and H2 variables.

all variables combined. The coefficients of the variables, as well as the *pseudo* - R^2 [85]³, are used to analyze each hypothesis.

As we can see in the results from H1, the Donor aid budget and the Public Grant have a positive coefficient, meaning that the Donor Aid (i.e., Spain in this case study) might influence the NGOs in the decision to ahead with a project in a country. However, the low R^2 indicates that these variables cannot properly explain the variance of the dependent variable and,

³ In R, McFadden's *pseudo* R^2 can be computed using the DescTools package.

therefore, this hypothesis cannot be validated. Regarding the constant, it is negative, which is to be expected because it is more likely to not implement a project than otherwise. Finally, it can be seen that Colonial History positively influences the output, a pattern repeated in each of the four models.

For H2, the coefficient for the GDP *per capita* shows a negative trend (i.e., this is the higher the GDP per capita, the less likely it is that a development project will be undertaken) and a highly statistically significant coefficient, as we might expect according to the value principled hypothesis. Regarding the UN LDC priority, the coefficient is negative, meaning that Spanish NGOs do not prioritize these countries (since these are the poorest countries from Africa and Asia, while Spanish NGOs prioritize Spanish-speaking countries). Finally, both the Africa Mission and Latin America Mission have a negative coefficient, an expected variable considering that these regional focused NGOs are smaller than other non-regional focused NGOs (e.g., Cruz Roja or Caritas) and, therefore, they develop fewer projects. Nevertheless, the low R^2 value of 0.16 still indicates that H2 cannot explain the variance in the dependent variable.

Finally, H3 seems to be the hypothesis that explains better the variance in the model, with a R^2 value higher than 0.5 (and a lower AIC, Akaike Information Criterion, indicating a more parsimonious model). Providing financial aid in the previous year (NGO budget previous year) to one country increases the probability of being prioritized by the organization. In addition, an NGO delegation has a positive coefficient, confirming that past experience in one country is important to understand NGO aid allocation.

Finally, the conclusions from H1, H2 and H3 can also be achieved when analyzing the fourth model that includes all the variables: the independent variables from the Back Donor Effect have a positive impact, all the variables regarding the principles and needs of the host have a negative impact, and the path dependence variables have positive impacts and are the most important variables according to their coefficients. In this case, the R^2 and the AIC are similar to H3, which reaffirms the importance of the NGO's past choices and experience to determine where the projects are developed.

Variance Inflation Factor of each independent variable

VARIABLES	VARIANCE INFLATION FACTOR
UN LDC	1.75
GDP per capita	1.87
Budget Previous Year	1.03
Delegation	1.04
Donor aid Budget	1.54
Colony	1.84
Latin America Mission	1.02
Africa Mission	1.02
Public Grant	1.06

Table 8.3: The results indicate that there exist no multicollinearity problem in this dataset, with very low VIF values that are never higher than 2.

Contingency Table (Budget Previous Year/Delegation, Project Developed)

	BUDGET PREVIOUS YEAR=0	BUDGET PREVIOUS YEAR>0
Project Developed=0	43439	1365
Project Developed=1	1417	3819

	DELEGATION=0	DELEGATION=1
Project Developed=0	44404	400
Project Developed=1	3308	1928

Table 8.4: As indicated by Spearman’s correlation in Figure 8.2, there exists a strong correlation between the variables from H3 and the target variable. However, this correlation is not enough to fully predict if a project is going to be developed in a country and, therefore, it is reason enough to discard the suspicion of data leakage.

Logistic Regression Models for H1, H2, H3

VARIABLES	H1: BACK DONOR EFFECT	H2: VALUES, PRINCIPLES AND NEEDS OF THE HOST OF THE COUNTRY	H3: PATH DEPENDENCE: NGO PAST CHOICES AND EXPERIENCE	INCLUDING ALL VARIABLES FROM H1, H2 and H3
<i>Constant</i>	-2.960497*** [0.022840]	1.78555*** [0.15046]	-3.669631*** [0.030106]	-0.955250*** [0.216968]
log(Donor Aid Budget)	0.053767*** [0.001761]			0.006622* [0.002750]
log(Public Grant)	0.177621*** [0.009132]			0.092552*** [0.011865]
log(GDP <i>per capita</i>)		-0.39960*** [0.01298]		-0.235970*** [0.018259]
UN LDCs		-0.13651** [0.04503]		-0.133233* [0.061525]
Africa Mission		-1.63304 *** [0.14246]		-1.440199*** [0.195621]
Latin America Mission		-0.22153*** [0.04448]		-0.206328** [0.063380]
log(Budget Previous Year)			0.230539*** [0.002823]	0.220699*** [0.002865]
NGO Delegation			2.617064*** [0.082455]	2.355460*** [0.084813]
Colonial history (<i>control variable</i>)	1.349853*** [0.037245]	2.46996*** [0.03803]	0.982788*** [0.047921]	1.184480*** [0.064942]
<i>R</i> ² and AIC				
Pseudo <i>R</i> ²	0.1712153	0.1595852	0.5017889	0.5144803
AIC	27807	28201	16719	16305

Table 8.5: Logistic Regression Models for H1, H2, H3, and the model that uses all the variables. For each independent variable we include Standard Error [between brackets]. We also include the significance of the variable. *significant at 10%, **significant at 5%, ***significant at 1%.

Explained LSTM model

Figure 8.3 includes two charts with the top 10 variables that have the most influence on the dependent variable according to the Shapley values. Indeed, the main conclusions from the regression analysis can be extrapolated in the LSTM model, i.e., the importance of the H3 variables is also shown in the LSTM model. However, the use of the LSTM allowed going further in the analysis, where the information from prior years (e.g., Donor Aid budget_2015 or Delegation_2015) indicates a path-dependence pattern beyond one year.

Most important independent variables according to the Shapley values

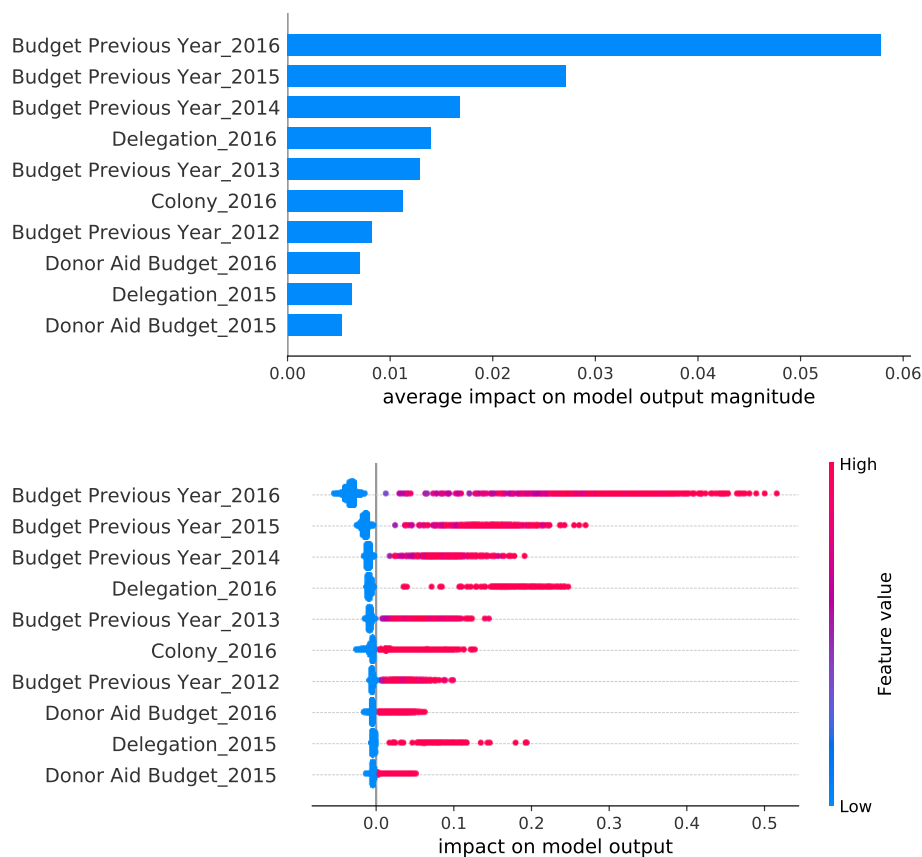


Figure 8.3: The most important independent variables according to the Shapley values' explanation. As indicated by the Regression Model, the Path Dependence variables are the most important independent variables in the model. However, LSTM allows us to include a temporal analysis absent in the Regression model, allowing us to determine historical Path Dependence variables (e.g., Budget_Previous_Year_2015 and Budget_Previous_Year_2014 highly influence the decision to develop a project in 2016).

Although it is evident that, at least for Spanish NGOs, the path-dependence hypothesis is confirmed, the paper also analyses the variables of the other two hypotheses (not included in 8.3 due to their low relevance), as can be seen in Figure 8.4. In spite of their negligible impact on model output, these features influence the output as expected. For instance, according to the UN LDCs’ Shapley values, Spanish NGOs do not prioritize the Least Developed Countries, since most of their projects are developed in former Spanish colonies, as previously seen in Figure 8.2. However, we see that a low GDP per capita increases the prediction. If we analyze the variables regarding the mission of the NGOs, we can see that those with a Latin America Mission slightly increase the probability of developing a project, while if we analyze the Africa Mission NGOs, we can see the opposite behavior. Finally, it is clear that the fact that an NGO has a Public Grant to develop a project should increase the model output.

Shapley values from other H1 and H2 variables

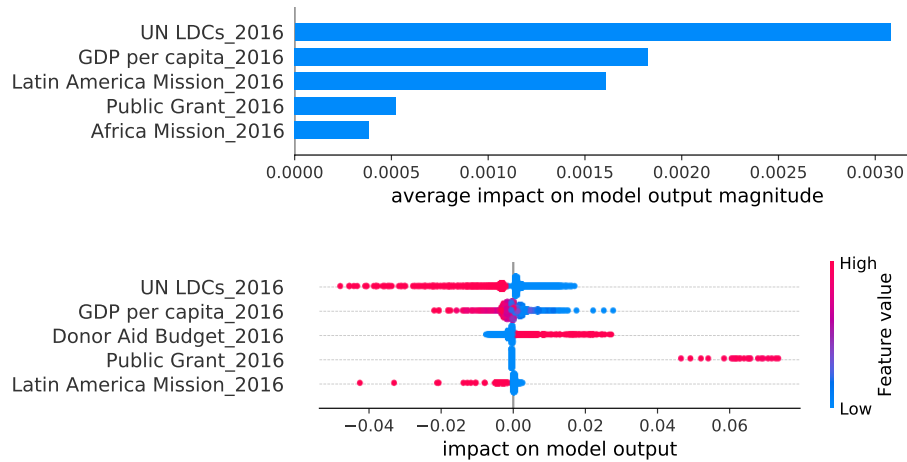


Figure 8.4: As we can see in these Figures, H1 and H2 variables are not the most relevant variables for the model, but are worthy of analysis. We would highlight that, according to the *UN LDCs* values, we can confirm that Spanish NGOs do not prioritize the countries from the UN LDC. If we analyze the variables regarding the mission of the NGOs, we can see that those with a Latin America Mission slightly increase the probability of developing a project, while if we analyze the Africa Mission NGOs we can see the opposite behavior. This is expected since most of the projects from Spanish NGOs are developed in Latin America. Finally, the models consider that the NGOs prioritizes countries with low GDP per capita, an expected behavior.

Closure

Within the international aid system, NGOs are key actors, as they are the ones that distribute most of the aid, both from public and private sources. Therefore, NGOs are one of the main focuses of attention when analyzing how such aid is prioritized. The question raised by previous studies is to what extent NGO aid follows a pragmatic or survival logic (developing projects in countries where they can receive donor aid), referred to in this paper as hypothesis 1 (H1) or whether, on the contrary, they move with a logic based on values and principles, more attached to the altruistic idea that surrounds the actions of this type of organization (referred to in this paper as H2). The work carried out in [95] proposes a third option: that NGOs develop projects in those countries where they have roots (referred to in the work as path dependence hypothesis or H3), either because they have developed projects in previous years or because they have a delegation.

These three hypotheses are analyzed in this paper using two data methods. The first method consists of developing several logistic regression models, where each model is intended to validate each of the above hypotheses. All the variables are significant, so it is not easy to evaluate which is the clearly determining logic. However, only the models with the path dependence independence variables achieve a high R2, around 0.5 (while the other models had R2 values close to 0.16), i.e. the information related to the existence of a project in the previous year or a Delegation is able to explain the variance of the dependent variable.

The second approach consists of fitting a Long-Short Term Memory Neural Network model, an Artificial Intelligence predictive algorithm that has persistence on historical data. This approach allows to determine if the path dependence patterns from H3 exist beyond one year (e.g. if a project developed before 2015 can influence the developing of a project in 2016). The explanation of the model obtained using Shapley values confirms the existence of this relationship, since among the ten most relevant variables to determine whether a project is developed in 2016, five correspond to whether a project existed in previous years (2011-2015)⁴, and two correspond to the existence of a Delegation in 2016 and 2015. The other three variables are if the country is a former colony of the back donor, and the Donor Aid Budget for that country in 2016 and 2015. This temporal perspective cannot be achieved through the classical regression approach, demonstrating that explained artificial intelligence models can be a great tool in social sciences, with non-restrictive fully interpretable data models.

⁴ As a reminder, the Budget Previous Year from a year X corresponds to the money spent in that country in the previous year, i.e. year X-1.

Part IV

Conclusions and Further Research

This work has previously explained three data science projects where the human factor has been a key element in their development, allowing us to understand, explore and apply the most important tools of explainable artificial intelligence. Below we will summarize the contributions of the work carried out. Then, we will draw some general conclusions about the current state of machine learning in general and interpretability tools in particular and analyze what it means to achieve a robust and transparent predictive model.

Chapter 9

Discussion

Undoubtedly, the use of predictive machine learning and other data science solutions is rapidly increasing, being applied in many fields beyond academic computing or technology companies, as we have explained in the introduction. We have found examples of this in this thesis, where we have implemented human-aware data science solutions in three different fields: utilities, banking and social sciences. We have analyzed existing solutions to achieve predictive models that are truly transparent and understandable to stakeholders (or researchers), implementing successful solutions in each case. In this section we draw some conclusions, considerations and thoughts.

9.1 This Work: Contributions and Results

The need to understand the predictions made by a predictive algorithm is not new, as the goal in machine learning has always been to achieve reliable predictions. However, during the last few years we have witnessed a boom in the need to understand such algorithms, with the emergence of new tools that address the need to respond to the concerns arising from the use of machine learning algorithms in our daily lives. This work pioneers the application of transparent machine learning solutions in different fields.

Our main contribution has been the development of the NTL detection system for Naturgy. This work has encompassed all machine learning processes, from data extraction to post-analysis. That said, our added value consisted in giving the system a layer of transparency to achieve high interpretability, which in general had not been analyzed in the literature: there exist some examples in which the explanations of the predictions are analyzed (e.g., [118]), but not with the depth that we offer in this thesis. Moreover, we highlight our focus on involving the stakeholder in the predictive process, since in many cases machine learning is regarded as a human substitute in business processes, which we do not consider to be true, due to the limitations outlined throughout the work.

The knowledge acquired in the NTL detection process was applied in

two other domains: predictive monitoring (business process management) and development cooperation (social science). In BPM we have been the first to implement highly explainable solutions to achieve transparent predictions in predictive monitoring. In development cooperation, we have also been pioneers implementing artificial intelligence solutions combined with Shapley values to understand how NGOs decide their projects, being also one of the first works implementing these techniques in social science.

9.2 General Discussion

9.2.1 The Importance of Data in Predictive Modeling

In general, data from machine learning projects can be divided between controlled data and observational data. Controlled data (also referred to in the literature as data by design and, depending on their structure, experimental data) adequately represent the actual scenario, i.e., controlled data are i.i.d. (independent and identically distributed) with respect to the test data set, have no bias (or, at least, the sampling bias is controlled by an expert) and, therefore, the process of building a model might not be as challenging as training a model using observational data. The problem with observational data is that they are often obtained for other purposes. The obvious consequence is that, in many cases, this results in biases and other data-related problems that might frequently go unnoticed. Despite this, most of the literature uses observational data. After all, it is quicker and cheaper to obtain because it does not need to be processed. In other words, the data are already "constructed" and, therefore, there is no need to analyze their correctness. Observational data are what they are.

This comment on observational data comes on the heels of the data used in the NTL detection system from Part II. As explained in this thesis, this work used the data from the company's visits that did not adequately represent customers as desired. Considering that this would be a problem, an optimal solution was put forward to generate exploration campaigns to obtain better customer representativeness, but this was not carried out due to the company's refusal. This caused certain blindness on the part of the data scientists that made it difficult for us to implement technical solutions, since by using biased data to validate our proposals we could never be entirely sure whether or not these changes were an improvement to the system.

Thus, explanatory algorithms (especially those that provide modular explanations), can be useful to compensate for problems related to the use of observational data. All problems relating to representativeness and biased patterns arise through model explanation, and for this reason our efforts in the NTL system evolved from the initial goal of building a very complex system with many variables to a more controlled environment in which the stakeholder controlled the system.

9.2.2 Involving the Stakeholder: Causality Validation and Benchmarking

The classical approach to hyperparameter tuning and model selection needs, as explained in Section 2.1.2, to use a validation dataset, with the goal of reproducing the unseen data set and thus determining how well the system generalizes. However, as we explain in Part II, when the labeled information is biased, the results in the validation dataset may not adequately represent how well our system generalizes. Thus, the classical approach of grid (or random) search for the optimal hyperparameter, and direct comparison of different algorithms on the same dataset, becomes obsolete or, at the very least, less reliable than one would want.

We explain how we analyze our NTL detection system based on learned patterns instead of classical benchmarking in different sections. For example, the man-in-the-loop approach explained in chapter 6 aims to implement a naive but functional method in which the stakeholder determines when the system is learning unwanted patterns and corrects it when possible. Auditing the algorithm beyond benchmarking through an in-depth explanation of the learned patterns allows us to detect biases and other unwanted behaviors that go unnoticed when using metrics on a dataset, especially when the i.i.d assumption is not met. Similarly, in the BPM example from Part III, we validate the explained LSTM approach because the explanations obtained are in line with the one obtained from human analysts.

Thus, the development of increasingly complex algorithms to exploit the vast amount of data available has not diminished the role of the human being, but has made it more indispensable than ever.

9.2.3 Interpretable vs Explained Black-Box Algorithms

This work's last and most important conclusion consists of the classical trade-off between interpretable algorithms and explained black-box algorithms. In general, most literature considers the latter option to provide more accuracy, while the interpretable algorithms approach should be reserved for simple predictive problems. This dichotomy should, at the very least, be seriously discussed based on our experience, as extensively explained in [117].

In general, we could agree that in a perfect scenario where the available data represents reality, with no biases or dataset shift, the black-box algorithm should be as accurate as the interpretable approach, since the black-box can reproduce the simpler patterns learned by the interpretable models. However, in real-world problems with messy data (due to biases and sub-optimal data generation processes) benchmarking can be misleading, and thus both interpretable and black-box algorithms could provide sub-optimal results, even though the latter can provide a priori better results in terms of benchmarking (e.g., due to overfitting) that may not gen-

eralize well on unseen data. The virtue of interpretable models is that, in general, the process of building a model is far more transparent and, therefore, it is easier to reprocess the data to represent reality and guarantee causal patterns properly.

The situation of having a sub-optimal system was faced during the development of the NTL detection system, and the introduction of Shapley values in the system provided us with an optimal method to explain Gradient Boosting Decision Tree models. In a sense, the original decision to use an Ensemble of Trees to predict the NTL cases allowed us to benefit from the existence of the Tree SHAP from [81], which provides robust and faithful explanations of what the system learned.

In any case, this thesis does not want to discourage the use of black-box algorithms, but rather advocates contextualizing their use in a context that requires it. The NGOs project from Part III is a good example where an explained black-box algorithm can provide a different point of view to the researcher, in such a way that it helps them to validate hypotheses that are difficult to do so with simpler algorithms.

Bibliography

- [1] Can machine learning build a better fico score? 6, 7
- [2] M. H. Afzali, M. Sunderland, S. Stewart, B. Masse, J. Seguin, N. Newton, M. Teesson, and P. Conrod. Machine-learning prediction of adolescent alcohol use: a cross-study, cross-cultural validation. *Addiction*, 114(4):662–671, 2019. 127
- [3] Agencia Estatal de Meteorología (España), Instituto de Meteorologia (Portugal). Climate atlas of the archipelagos of The Canary Islands, Madeira and The Azores. http://www.aemet.es/documentos/es/conocerlas/recursos_en_linea/publicaciones_y_estudios/publicaciones/2Atlas_climatologico/Atlas_Clima_Macaronesia__Baja.pdf. 41
- [4] Agencia Estatal de Meteorología (España), Instituto de Meteorologia (Portugal). Iberian climate atlas. http://www.aemet.es/documentos/es/conocerlas/recursos_en_linea/publicaciones_y_estudios/publicaciones/Atlas-climatologico/Atlas.pdf. 41
- [5] A. Alin. Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):370–374, 2010. 13
- [6] N. S. Altman. An introduction to kernel and nearest-neighbor non-parametric regression. *The American Statistician*, 46(3):175–185, 1992. 23
- [7] D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018. 73
- [8] C. Amrit, T. Paauw, R. Aly, and M. Lavric. Identifying child abuse through text mining and machine learning. *Expert systems with applications*, 88:402–418, 2017. 127
- [9] R. M. Anderson, H. Heesterbeek, D. Klinkenberg, and T. D. Hollingsworth. How will country-based mitigation measures influence the course of the covid-19 epidemic? *The Lancet*, 395(10228):931–934, 2020. 4

- [10] E. W. S. Angelos, O. R. Saavedra, O. A. C. Cortés, and A. N. de Souza. Detection and identification of abnormalities in customer consumptions in power distribution systems. *IEEE Transactions on Power Delivery*, 26(4):2436–2442, Oct 2011. [38](#)
- [11] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. [65](#), [109](#)
- [12] V. Badrinath Krishna, G. A. Weaver, and W. H. Sanders. Pca-based method for detecting integrity attacks on advanced metering infrastructure. In J. Campos and B. R. Haverkort, editors, *Quantitative Evaluation of Systems*, pages 70–85, Cham, 2015. Springer International Publishing. [38](#)
- [13] L. Beretta and A. Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):197–208, 2016. [13](#)
- [14] M. R. Berthold, N. Cebon, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007. [48](#)
- [15] M. Bevilacqua, F. Ciarapica, and C. Paciarotti. Business process reengineering of emergency management procedures: A case study. *Safety science*, 50(5):1368–1376, 2012. [114](#)
- [16] J. M. Bland and D. G. Altman. The odds ratio. *Bmj*, 320(7247):1468, 2000. [87](#)
- [17] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. [26](#)
- [18] L. Breiman. Out-of-bag estimation. 1996. [16](#)
- [19] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. [26](#)
- [20] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Routledge, 2017. [23](#)
- [21] D. Breuker, P. Delfmann, M. Matzner, and J. Becker. Designing and evaluating an interpretable predictive modeling technique for business processes. In *International Conference on Business Process Management*, pages 541–553. Springer, 2014. [119](#)

- [22] T. Büthe, S. Major, and A. d. M. e Souza. The politics of private foreign aid: humanitarian principles, economic development objectives, and organizational interests in ngo private aid allocation. *International organization*, 66(4):571–607, 2012. [128](#)
- [23] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito. Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2018. [37](#)
- [24] J. E. Cabral, J. O. Pinto, E. M. Martins, and A. M. Pinto. Fraud detection in high voltage electricity consumers using data mining. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, pages 1–5. IEEE, 2008. [38](#)
- [25] A. Calvo, B. Coma-Puig, J. Carmona, and M. Arias. Knowledge-based segmentation to improve accuracy and explainability in non-technical losses detection. *Energies*, 13(21):5674, 2020. [11](#), [35](#)
- [26] J. Chen and J. Shao. Nearest neighbor imputation for survey data. *Journal of official statistics*, 16(2):113, 2000. [13](#)
- [27] S.-J. Chen, T.-S. Zhan, C.-H. Huang, J.-L. Chen, and C.-H. Lin. Nontechnical loss and outage detection using fractional-order self-synchronization error-based fuzzy petri nets in micro-distribution systems. *IEEE Transactions on smart grid*, 6(1):411–420, 2015. [38](#)
- [28] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 785–794, New York, NY, USA, 2016. ACM. [26](#)
- [29] W. W. Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995. [23](#)
- [30] B. Coma-Puig, A. Calvo, J. Carmona, and R. Gavaldà. A case study of improving a non-technical losses detection system through explainability. submitted: Special Issue on Explainable and Interpretable Machine Learning and Data Mining (Data Mining and Knowledge Discovery). [10](#), [35](#), [60](#), [85](#)
- [31] B. Coma-Puig and J. Carmona. A quality control method for fraud detection on utility customers without an active contract. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC '18*, pages 495–498, New York, NY, USA, 2018. ACM. [9](#), [35](#), [73](#), [82](#), [91](#), [92](#), [95](#)
- [32] B. Coma-Puig and J. Carmona. Bridging the gap between energy consumption and distribution through non-technical loss detection. *Energies*, 12(9), 2019. [10](#), [35](#), [38](#), [51](#), [63](#), [66](#)

- [33] B. Coma-Puig and J. Carmona. A human-in-the-loop approach based on explainability to improve ntl detection. *arXiv preprint arXiv:2009.13437*, 2020. [10](#)
- [34] B. Coma-Puig and J. Carmona. Non-technical losses detection in energy consumption focusing on energy recovery and explainability. *Machine Learning*, pages 1–31, 2021. [10](#), [35](#), [98](#)
- [35] B. Coma-Puig and J. Carmona. Non-technical losses detection in energy consumption focusing on energy recovery and explainability. *Machine Learning*, 2021. [95](#)
- [36] B. Coma-Puig, J. Carmona, R. Gavaldà, S. Alcoverro, and V. Martín. Fraud detection in energy consumption: A supervised approach. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 120–129. IEEE, 2016. [9](#), [18](#), [19](#), [35](#), [38](#), [39](#), [51](#), [66](#)
- [37] B. C. Costa, B. L. Alberto, A. M. Portela, W. Maduro, and E. O. Eler. Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. *International Journal of Artificial Intelligence & Applications*, 4(6):17, 2013. [37](#)
- [38] A. D’Amato. Can/should computers replace judges? *Georgia Law Review*, 11:11–36, 1977. [5](#)
- [39] S. Dandl, C. Molnar, M. Binder, and B. Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020. [32](#)
- [40] T. Davenport and R. Kalakota. The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94, 2019. [5](#)
- [41] 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, New York, NY, USA, 2006. ACM. [18](#)
- [42] 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, New York, NY, USA, 2006. ACM. [63](#)
- [43] M. De Leoni, W. M. van der Aalst, and M. Dees. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56:235–257, 2016. [117](#)

- [44] C. Di Francescomarino, M. Dumas, F. M. Maggi, and I. Teinemaa. Clustering-based predictive process monitoring. *IEEE transactions on services computing*, 12(6):896–909, 2016. [118](#)
- [45] T. E. Dielman. *Applied regression analysis for business and economics*. Duxbury/Thomson Learning Pacific Grove, CA, 2001. [23](#)
- [46] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. [125](#)
- [47] C. Drummond and R. C. Holte. Explicitly representing expected cost: An alternative to roc representation. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 198–207, 2000. [18](#)
- [48] C. Drummond and N. Japkowicz. Warning: statistical benchmarking is addictive. kicking the habit in machine learning. *Journal of Experimental & Theoretical Artificial Intelligence*, 22(1):67–80, 2010. [21](#), [63](#)
- [49] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, et al. *Fundamentals of business process management*, volume 1. Springer, 2013. [114](#), [115](#)
- [50] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the international conference on machine learning*, volume 28. ACM New York, USA, 2013. [4](#)
- [51] E. A. Feigenbaum, B. G. Buchanan, and J. Lederberg. On generality and problem solving: A case study using the dendral program. 1970. [3](#)
- [52] G. Fenza, M. Gallo, and V. Loia. Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, 7:9645–9657, 2019. [37](#)
- [53] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936. [25](#)
- [54] F. Folino, M. Guarascio, and L. Pontieri. A prediction framework for proactively monitoring aggregate process-performance indicators. In *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, pages 128–133. IEEE, 2015. [118](#)
- [55] V. Ford, A. Siraj, and W. Eberle. Smart grid energy fraud detection using artificial neural networks. In *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, pages 1–6, Dec 2014. [37](#)

- [56] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. [26](#)
- [57] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. [32](#)
- [58] R. Galanti, B. Coma-Puig, M. de Leoni, J. Carmona, and N. Navarin. Explainable predictive process monitoring. In *2020 2nd International Conference on Process Mining (ICPM)*, pages 1–8. IEEE, 2020. [8](#), [11](#), [113](#), [119](#), [120](#), [125](#)
- [59] S. García, J. Luengo, and F. Herrera. *Data preprocessing in data mining*, volume 72. Springer, 2015. [12](#)
- [60] S. Ghosh and D. L. Reilly. Credit card fraud detection with a neural-network. volume 3, pages 621–630. IEEE, 1994. [5](#)
- [61] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger. The challenge of non-technical loss detection using artificial intelligence: A survey. *International Journal of Computational Intelligence Systems*, 10(1):760, 2017. [8](#)
- [62] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger. The challenge of non-technical loss detection using artificial intelligence: A survey. *International Journal of Computational Intelligence Systems*, 10:760–775, 2017/01. [38](#)
- [63] K. Godfrey. Simple linear regression in medical research. *New England Journal of Medicine*, 313(26):1629–1636, 1985. [23](#)
- [64] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. [22](#)
- [65] B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*, 2018. [32](#)
- [66] D. Gunning and D. Aha. Darpa’s explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, 2019. [22](#), [28](#)
- [67] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pages 986–996. Springer, 2003. [23](#)
- [68] M. A. Hanif, F. Khalid, and M. Shafique. Cann: Curable approximations for high-performance deep neural network accelerators. In

- 2019 56th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2019. [28](#)
- [69] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993. [23](#)
- [70] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. [19](#), [69](#), [99](#)
- [71] P. Kadurek, J. Blom, J. Cobben, and W. L. Kling. Theft detection and smart metering practices and expectations in the netherlands. In *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, pages 1–6. IEEE, 2010. [38](#)
- [72] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):15, 2012. [73](#)
- [73] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017. [26](#)
- [74] J.-H. Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, 53(11):3735–3745, 2009. [16](#)
- [75] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995. [16](#)
- [76] V. B. Krishna, R. K. Iyer, and W. H. Sanders. Arima-based modeling and validation of consumption readings in power grids. In *International Conference on Critical Information Infrastructures Security*, pages 199–210. Springer, 2015. [84](#)
- [77] J. Kruppa, A. Schwarz, G. Armingier, and A. Ziegler. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40(13):5125–5131, 2013. [5](#)
- [78] P. Küng and C. Hagen. The fruits of business process management: an experience report from a swiss bank. *Business process management journal*, 2007. [115](#)
- [79] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, and F. M. Maggi. Complex symbolic sequence encodings for predictive

- monitoring of business processes. In *International Conference on Business Process Management*, pages 297–313. Springer, 2016. 117
- [80] Y. Liu and S. Hu. Cyberthreat analysis and detection for energy theft in social networking of smart homes. *IEEE Transactions on Computational Social Systems*, 2(4):148–158, 2015. 38
- [81] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018. 73, 93, 146
- [82] R. S. Mans, M. Schonenberg, M. Song, W. M. van der Aalst, and P. J. Bakker. Application of process mining in healthcare—a case study in a dutch hospital. In *International joint conference on biomedical engineering systems and technologies*, pages 425–438. Springer, 2008. 4
- [83] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2145–2148, 2020. 21
- [84] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés. Predictive monitoring of business processes: A survey. *IEEE Transaction on Services Computing*, 11(6):962–977, 2018. 119
- [85] D. McFadden et al. Conditional logit analysis of qualitative choice behavior. 1973. 134
- [86] G. M. Messinis and N. D. Hatziargyriou. Review of non-technical loss detection methods. *Electric Power Systems Research*, 158:250–266, 2018. 38
- [87] C. Molnar. *Interpretable Machine Learning*. 2019. <https://christophm.github.io/interpretable-ml-book/>. 94
- [88] I. Monedero, F. Biscarri, C. León, J. I. Guerrero, J. Biscarri, and R. Millán. Detection of frauds and other non-technical losses in a power utility using pearson coefficient, bayesian networks and decision trees. *International Journal of Electrical Power & Energy Systems*, 34(1):90 – 98, 2012. 87
- [89] K. P. Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18(60):1–8, 2006. 23
- [90] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad. Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE transactions on Power Delivery*, 25(2):1162–1171, 2009. 37

- [91] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and F. Nagi. Improving svm-based nontechnical loss detection in power utility using the fuzzy inference system. *IEEE Transactions on Power Delivery*, 26(2):1284–1285, April 2011. [37](#)
- [92] A. Nait Aicha, G. Englebienne, K. S. Van Schooten, M. Pijnappels, and B. Kröse. Deep learning to predict falls in older adults based on daily-life trunk accelerometry. *Sensors*, 18(5):1654, 2018. [4](#)
- [93] N. Navarin, B. Vincenzi, M. Polato, and A. Sperduti. LSTM networks for data-aware remaining time prediction of business process instances. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2017)*, 2017. [117](#), [118](#), [119](#)
- [94] I. Nunes and D. Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3–5):393–444, Dec. 2017. [125](#)
- [95] N. Orue, L. Muñoz, and B. Coma-Puig. The logic behind ngos’ aid allocation: a complex choice based on past decisions. Submitted. [8](#), [11](#), [113](#), [128](#), [133](#), [140](#)
- [96] J. P. Papa, A. X. Falcao, and C. T. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009. [38](#)
- [97] G. Park and M. Song. Prediction-based resource allocation using lstm and minimum cost and maximum flow algorithm. In *Proceedings of the International Conference on Process Mining (ICPM 2019)*, pages 121–128, 2019. [119](#)
- [98] J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1):259–268, 2015. [5](#)
- [99] J. Pearl. *Causality*. Cambridge university press, 2009. [29](#), [65](#)
- [100] J. Pearl and D. Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018. [21](#), [29](#), [65](#)
- [101] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. [30](#), [48](#)

- [102] L. A. M. Pereira, L. C. S. Afonso, J. P. Papa, Z. A. Vale, C. C. O. Ramos, D. S. Gastaldello, and A. N. Souza. Multilayer perceptron neural networks training through charged system search and its application for non-technical losses detection. In *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, pages 1–6, April 2013. [37](#)
- [103] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features, 2017. [26](#), [30](#)
- [104] F. Provost and T. Fawcett. Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59, 2013. [3](#)
- [105] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. [21](#), [73](#)
- [106] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018. [4](#)
- [107] C. C. O. Ramos, A. N. de Sousa, J. P. Papa, and A. X. Falcao. A new approach for nontechnical losses detection based on optimum-path forest. *IEEE Transactions on Power Systems*, 26(1):181–189, Feb 2011. [37](#)
- [108] C. C. O. Ramos, D. Rodrigues, A. N. de Souza, and J. P. Papa. On the study of commercial losses in brazil: A binary black hole algorithm for theft characterization. *IEEE Transactions on Smart Grid*, 9(2):676–683, March 2018. [37](#)
- [109] J.-R. Rehse, N. Mehdiyev, and P. Fettke. Towards explainable process predictions for industry 4.0 in the dfki-smart-lego-factory. *KI-Künstliche Intelligenz*, 33(2):181–187, 2019. [119](#)
- [110] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016. [29](#)
- [111] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016. [31](#)

- [112] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 32
- [113] M. Robila and S. A. Robila. Applications of artificial intelligence methodologies to behavioral and social sciences. *Journal of Child and Family Studies*, 29(10):2954–2966, 2020. 127
- [114] C. Ross and I. Swetlitz. Ibm’s watson supercomputer recommended ‘unsafe and incorrect’ cancer treatments, internal documents show. *Stat News* <https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments>, 2018. 6
- [115] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, and K. Sricharan. Recognizing abnormal heart sounds using deep learning. *arXiv preprint arXiv:1707.04642*, 2017. 4
- [116] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 15
- [117] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 145
- [118] M. Salman Saeed, M. W. Mustafa, U. U. Sheikh, T. A. Jumani, I. Khan, S. Atawneh, and N. N. Hamadneh. An efficient boosted c5. 0 decision-tree-based classification approach for detecting non-technical losses in power utilities. *Energies*, 13(12):3242, 2020. 143
- [119] Y. Sasaki. The truth of the f-measure. *Teach Tutor Mater*, 01 2007. 17
- [120] A. Senderovich, M. Weidlich, A. Gal, and A. Mandelbaum. Queue mining for delay prediction in multi-class service processes. *Information Systems*, 53:278–295, 2015. 117
- [121] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953. 31, 94
- [122] N. Shin and D. F. Jemella. Business process reengineering and performance improvement: The case of chase manhattan bank. *Business Process Management Journal*, 2002. 115
- [123] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014. 13
- [124] E. H. Shortliffe. Mycin: a rule-based computer program for advising physicians regarding antimicrobial therapy selection. Technical report, STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE, 1974. 3

- [125] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017. [32](#)
- [126] R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *arXiv preprint arXiv:2106.03253*, 2021. [57](#)
- [127] J. V. Spirić, M. B. Dočić, and S. S. Stanković. Fraud detection in registered electricity time series. *International Journal of Electrical Power & Energy Systems*, 71:42–50, 2015. [38](#)
- [128] E. Strickland. Ibm watson, heal thyself: How ibm overpromised and underdelivered on ai health care. *IEEE Spectrum*, 56(4):24–31, 2019. [6](#)
- [129] E. Sulis and A. Di Leva. An agent-based model of a business process: The use case of a hospital emergency department. In *International Conference on Business Process Management*, pages 124–132. Springer, 2017. [114](#)
- [130] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. [32](#)
- [131] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. [32](#)
- [132] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas. Predictive business process monitoring with LSTM neural networks. In *Proceedings of 29th International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, pages 477–492, 2017. [118](#), [119](#)
- [133] I. Teinemaa, M. Dumas, F. M. Maggi, and C. Di Francescomarino. Predictive business process monitoring with structured and unstructured data. In *International Conference on Business Process Management*, pages 401–417. Springer, 2016. [118](#)
- [134] W. Van Der Aalst. Data science in action. In *Process mining*, pages 3–23. Springer, 2016. [3](#), [4](#)
- [135] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, and C. Di Francescomarino. Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring. In *International Conference on Business Process Management*, pages 218–229. Springer, 2016. [117](#)

- [136] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pages 758–770. Springer, 2005. 13
- [137] J. Vom Brocke and J. Mendling. *Digital Innovation and Business Transformation in Practice*. Berlin et al.: Springer, pages 244–253, 2018. 114
- [138] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017. 32
- [139] J. Wainer and G. Cawley. Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, 182:115222, 2021. 16
- [140] D. P. Wall, R. Dally, R. Luyster, J.-Y. Jung, and T. F. DeLuca. Use of artificial intelligence to shorten the behavioral diagnosis of autism. 2012. 127
- [141] W. Wang, I. Hernandez, D. A. Newman, J. He, and J. Bian. Twitter analysis: Studying us weekly trends in work stress and emotion. *Applied Psychology*, 65(2):355–378, 2016. 127
- [142] Y. Wang, Q. Chen, D. Gan, J. Yang, D. S. Kirschen, and C. Kang. Deep learning-based socio-demographic information identification from smart meter data. *IEEE Transactions on Smart Grid*, 10(3):2593–2602, 2018. 37
- [143] Z. Xiao, Y. Xiao, and D. H.-C. Du. Exploring malicious meter inspection in neighborhood area smart grids. *IEEE Transactions on Smart Grid*, 4(1):214–226, 2013. 38
- [144] A. Xu. Chinese judicial justice on the cloud: a future call or a pandora’s box? an analysis of the ‘intelligent court system’ of china. *Information & Communications Technology Law*, 26(1):59–71, 2017. 5
- [145] Z. Zhang, G. Mayer, Y. Dauvilliers, G. Plazzi, F. Pizza, R. Fronczek, J. Santamaria, M. Partinen, S. Overeem, R. Peraita-Adrados, et al. Exploring the clinical features of narcolepsy type 1 versus narcolepsy type 2 from european narcolepsy network database with machine learning. *Scientific reports*, 8(1):1–11, 2018. 27