INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Modernising Customer Service in Retail: A Worten Case Study on Automated Complaint Classification

Inês Rodrigues Casimiro

Master in Data Science

Supervisor:
PhD Anabela Ribeiro Dias da Costa, Assistant Professor,
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor
PhD Diana Elisabeta Aldea Mendes, Associate Professor,
ISCTE - Instituto Universitário de Lisboa

October, 2023

BUSINESS
SCHOOL

TECNOLOGIAS
E ARQUITETURA

Department of Quantitative Methods for Management and Economics

Department of Information Science and Technology

Modernising Customer Service in Retail: A Worten Case Study on Automated Complaint Classification

Inês Rodrigues Casimiro

Master in Data Science

Supervisor:
PhD Anabela Ribeiro Dias da Costa, Assistant Professor,
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Diana Elisabeta Aldea Mendes, Associate Professor,
ISCTE - Instituto Universitário de Lisboa

October, 2023

Acknowledgments

To Professor Anabela Costa and Professor Diana Mendes for accepting to be the supervisors of this dissertation and for guiding me along this path, which marks the culmination of my academic journey. Your support, availability and generosity were crucial in the development of this project.

To all the Teaching and Non-teaching staff at ISCTE-IUL who have cooperated in my learning and growth during this Master's degree.

To my Parents, who have support me in every decision. Who have always believed in me and gave me all the tools I needed to thrive. And to my Brothers, for the challenging but very rewarding journey it has been, to grow up alongside them.

To Guilherme for being the most generous person. Your words of motivation and constant support made this possible.

To the rest of my Family and Friends, for making my achievements their own and for their friendship and affection that I value so much.

To all the people I have come across at Worten for their teachings and especially to my Team for welcoming me so well and for the feeling of mutual help that is always present.

Resumo

O aparecimento de retalhistas com capacidade de entregar produtos no próprio dia e a preços muito competitivos, como a Amazon, levou a um aumento de expectativas por parte dos clientes. Quando a qualidade do serviço praticado fica aquém do esperado, os clientes recorrem a reclamações para demonstrar o seu descontentamento, e é do interesse dos retalhistas resolver o problema o mais rápido possível para evitar perder clientes.

Uma vez que o processo de análise de reclamações consome bastante tempo, este estudo visa propor um método de classificar as reclamações endereçadas à Worten, de forma automática. Assim, foram realizadas dezasseis experiências com oito algoritmos de *Machine Learning* (ML) diferentes, seguindo a metodologia *Cross Industry Standard Process for Data Mining* (CRISP-DM). As experiências efetuadas compreenderam a redução do número de classes, modelos de *Transfer Learning*, diferentes tipos de balanceamento de classes, entre outros.

O modelo *Support Vector Machine* (SVM) obteve a melhor classificação, com uma Acurácia de 71,41%, na experiência em que foram eliminadas as três classes mais difusas das seis classes originais (*Time*, *Technical Problem*, *Client*, *Money*, *Service* e *Other*).

Palavras-chave: Worten; Retalho de Eletrónica; Reclamações; *Machine Learning*; Processamento de Linguagem Natural; Classificação de Texto.

Abstract

The emergence of retailers able to deliver products on the same day and at very competitive prices, such as Amazon, has caused customers to raise their expectations. When the quality of service falls short of the expected, customers resort to complaints to show their dissatisfaction, and it is in the retailers' interest to resolve the problem as quickly as possible to avoid losing customers.

Since the process of analysing complaints is very time-consuming, this study aims to propose a method for classifying the complaints addressed to Worten, automatically. Thus, sixteen experiments were performed with eight different Machine Learning (ML) algorithms, following the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. The experiments included reducing the number of classes, Transfer Learning models, and different types of class balancing, among others.

The Support Vector Machine (SVM) model obtained the best classification, with an Accuracy of 71.41%, in the experiment in which the three most diffuse of the six original classes (*Time*, *Technical Problem*, *Client*, *Money*, *Service and Other*) were eliminated.

Keywords: Worten; Electronics Retail; Complaints; Machine Learning; Natural Language Processing; Text Classification.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

ATM	Automated Teller Machine
BERT	Bidirectional Encoder Representations from Transformers
BN	Bayes Networks
BNS	Bi-Normal Separation
BoW	Bag of Words
CHI	Chi Square
CRISP-DM	Cross Industry Standard Process for Data Mining
CV	Cross-Validation
DECO	*Associação Portuguesa para a Defesa do Consumidor*
DT	Decision Tree
HTML	HyperText Markup Language
ICF	Inverse Class Frequency
IG	Information Gain
IT	Information Technology
kNN	k-Nearest Neighbor
LR	Logistic Regression
ML	Machine Learning
MNB	Multinomial Naïve Bayes
NLP	Natural Language Processing
OVO	One-vs-One
OVR	One-vs-Rest
RF	Random Forest
SL	SimpleLogistic
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TCW	Term Class Weight
TCW-ICF	Term Class Weight-Inverse Class Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
WLLR	Weighted Log Likelihood Ratio

Introduction

Customer dissatisfaction with the shopping experience in retail companies is expressed in the form of complaints through various channels. With the rise of social media, customers have started to use this channel to voice their complaints, which can have a significant impact on the company's image (Miquel-Romero *et al.* 2020). Complaints are indicative of discontent, but they are also a form of feedback (Vollero *et al*., 2021). This allows the retailer to fix the poor service, assuring customer satisfaction and retention (Jeanpert *et al*., 2021). Thus, it is essential that complaints are dealt with as promptly and effectively as possible to ensure ties with customers (Frasquet *et al*., 2019).

The main focus of this research is to propose a way of automatically classifying the Portuguese complaints received by Worten according to their subject: whether they are related to technical problems, delays in delivery, unavailability of different payment methods, information provided, or quality of service, for instance. A task that takes days to complete could be reduced to just a few hours with the aim of ML algorithms, bearing in mind that human validation will be required.

The methodology adopted, CRISP-DM, is the most commonly used for Data Science projects and consists of six stages: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. Each phase can be revisited more than once, and those that are unnecessary can be skipped over (Martínez-Plumed *et al*., 2021).

This study consists of five Chapters, following the path of the CRISP-DM methodology, the first of which contextualizes the problem and presents articles of reference for automating routine tasks as those at Worten, a well-known electronics retailer in Portugal. The variety of ML models used in similar case studies is evident, as are the preprocessing and feature extraction techniques. The company is briefly introduced as well as the Sonae group, of which it is a part.

Chapter 2 covers the Data Preparation stage, from eliminating complaints that did not fit to being classified, as they were a transcript of the complaint made by the customer, to the preprocessing applied. A total of seven techniques were employed, including the

removal of hyperlinks, stop words and lemmatization, among others. The handling of missing values and complaints that are not written in Portuguese is addressed.

In order to better understand the data under study, a variety of graphs were created and analysed in Chapter 3. Firstly, for each attribute, clusters were formed with the most relevant keywords from the complaints, and then based on the number of words in each complaint. To perceive the distribution of the values for different attributes, bar charts were generated. In this Chapter is defined the strategy that led to the creation of six classes (*Time*, *Technical Problem*, *Client*, *Money*, *Service and Other*) to automatically classify complaints according to their topic, and the most common words in each class were represented in pie charts.

The fourth stage of CRISP-DM, Modelling, is approached in Chapter 4. The eight ML classifiers used: Decision Tree (DT), Support Vector Machine, Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Random Forest (RF), XGBoost, LightGBM, Catboost and Bidirectional Encoder Representations from Transformers (BERT) are explained. Each of the sixteen experiments performed, grouped into six Cases, is detailed, which include the classification of six classes with different types of class balancing, an experiment with no preprocessing applied, and another considering only the most common words in each class. Subsquently, the number of classes was halved, an experiment was carried out with a Transfer Learning model and the last one with redefined and more targeted classes.

The last Chapter presents and analyses the results of each Case and the different Cases are compared with each other. At the end, a final appreciation is made and the next steps to be implemented are defined.

CHAPTER 1

Literature Review

Digitalization is now the core of our society since it is present in all fields, from science to economy to the arts, and it is inevitably part of our daily lives. People not only use digital technology to work but also to connect with each other via their computers, tablets, or smartphones. Digitalization is assumed to be the progressive transition from real to digital (Bowen & Giannini, 2014). Retailers also followed the evolution as they kept growing. That gave them better knowledge about their sales, response times, and so on (Watson, 2011).

The fact that society, and thus customers, use their technology devices during the entire purchasing experience raises many challenges for retailers, such as competing in a global market and e-commerce (Arkenback, 2019). As part of digitalization, e-commerce includes converting physical products into digital services, customers' suggestions in social media, and integrating electronic devices into the shopping process. The proliferation of online stores was one of the earliest effects of digitalization in Retail, endorsing a possible extension of offers (Hagberg *et al*., 2016). E-commerce is a form of retailing using the Internet. Retail companies make their products available on the website and accessible to all customers with Internet access. Products purchased online are shipped to customers directly by the Internet, while logistics delivers products purchased in physical stores.

Retailers should set their priorities to meet and exceed customer expectations, and with the spread of online shopping, arose the need to adapt their business. One of the advantages of e-commerce is that customers are able to buy and order products unavailable in physical stores.

For a retailer to be competitive, it is important to establish strategies to acquire new clients, not forgetting the old ones (Hooda, 2011), and it depends on how they adapt to the outer operating environment (Ramazanov *et al*., 2021). Amazon has been the e-commerce leader since 1995, providing the best e-retail globally (Sadq *et al*., 2018) and is presumed guilty for the decrease in sales in physical stores due to their lower prices, enormous product variety, outstanding customer service, fast shipping and return policy. Amazon's innovation has changed how customers interact with other retailers and its impact on increasing customer expectations is known as the "Amazon effect", or simply "Amazonification". The customer expectations standards were set so high by Amazon that other retailers are facing a disadvantage, as consumers are now less satisfied with their services, comparing them to

Amazon's. When customers' expectations are not fulfilled, discontent is shown in the form of complaints (Vollero *et al*., 2021).

From the company's perspective, it gives them key information on what is going wrong and where they can perform better. Complaints are a critical indicator of inadequate performance. Therefore, must be analysed from a constructive point of view. The advantages of doing so include the measurement of the company's performance, a better knowledge of their customers resulting in an approximation with the company, and it is also a way of getting involuntary feedback (E. K. Özyirmidokuz & M. H. Özyirmidokuz, 2014).

On the other hand, online complaints can also be a key factor for a customer. Compared with positive reviews, complaints have the most influence and can play a decisive role in persuading the customer to buy or not the product. In fact, it is the first thing most people look at.

The sooner the complaints are addressed, the sooner the problem will be solved, and for this to happen, it is necessary to reduce the workload of the employees who oversee this task (Behere *et al.*, 2020). Supervised Machine Learning techniques involving Natural Language Processing (NLP), namely Text Classification, are a proven way to do it (Naresh *et al.*, 2019). A faster response time will improve customer satisfaction (Omurca *et al*., 2021), not damaging the company's image through word-of-mouth, nor possibly affecting its sales and market share (Lee & Cranage, 2014).

Sonae is a multinational company and the largest private employer in Portugal, operating in more than 60 countries across all continents. It manages several business sectors: financial services, technology, real estate, telecommunications, and retail.

Worten, the market leader in the business segment, controls Sonae's specialized electronics retail. Worten has over 240 stores in Portugal and Spain and the most prominent Portuguese e-commerce website, extending its offer to over six million products. In addition to household appliances and Information Technology (IT) products, Worten is also a leader in the area of repairs under the Worten Resolve seal and, most recently, has started to focus on offering wellness, decoration, and pet products, among others.

Self-titled a digital company, with physical stores and a human touch, Worten aims to take advantage of new technologies to boost its business. Receiving thousands of complaints every year, classifying and handling them becomes a challenge. This study proposes an automatic classification of complaints, according to its topic, through Machine Learning algorithms.

In order to analyse recent studies of this domain, the collection of relevant literature was mandatory. For this purpose, two search engines were used: Google Scholar and Scopus, two

of the three most relevant citation databases used for academic ends (Harzing & Alakangas, 2015). Studies show that Google Scholar covers more articles for most subjects, but Scopus does not provide very distant results in what comes to Journal Publications or Conference Papers in Engineering and Computer Science (Martín-Martín *et al.*, 2018).

The query chosen to narrow the field of research was *((("electronics retail company" OR "consumer electronic company") AND "text complaints") AND ("text mining" OR "classification" OR "machine learning"))*, to understand if there were any studies related to the analysis of customer complaints in electronics retail companies using any text mining techniques, but no results were shown.

Then, another query was applied *("text complaints" AND "machine learning")* to gather studies that involved complaints in text form and machine learning algorithms, across all fields and not just retail. Still, only one result appeared, indicating that the search should be broader.

Given the previous outcomes, the query *(("complaints" AND "supervised learning") OR ("complaints" AND "multi label classification"))* was used and resulted in 36 Articles and Conference Papers published after 2017, of which 12 were selected. By analysing the title, the keywords and the abstract of the articles shown, the ones that were not directly related to the study and those unavailable for free consultation were excluded.

Subsequently, in an attempt to understand if Transfer Learning approaches existed, the query chosen was ("transfer learning" AND "supervised learning" AND "text classification"), which returned 13 significant articles out of a total of 22 articles published in 2017 and onwards. In addition to the exclusion criteria applied in the previous query, Conference and systematic reviews were also excluded. Thus, 26 articles will be considered in total, and it is important to note that only one of the articles dealt with text written in Portuguese.

Lastly, to perceive how digitalization revolutionized retail companies through e-commerce, the query ("retail company" AND "digitalization") was applied and resulted in 10 articles to be reviewed.

The articles collected for this analysis were published in different Journals and Conferences from 2017 to 2023 (Table 1), with 2022 predominating with the most articles published.

Table 1 - Publication sites of selected articles.

Publication Site	**No. of Articles**
Conference Paper	20
Article	16

Table 2 presents the preprocessing techniques used across the gathered articles. Stop words removal is the preprocessing technique most used among all articles analysed, and it consists of eliminating prepositions and conjunctions with no semantic meaning (Naresh *et al*., 2019). Followed by the most popular, is special characters removal, covering HyperText Markup Language (HTML) tags (Singh *et al*., 2020), punctuation and numbers (Lee *et al*., 2019), Stemming that transforms words into its lemma and affixes (Fahrudin *et al*., 2019), and Tokenization.

On the other hand, data auto-correction, which aims to correct misspelled words through regular expressions (Guru *et al*., 2018) and the change from uppercase to lowercase words (Lee *et al*., 2019) were the least preprocessing techniques used by the authors of the articles in study, as well as Lemmatization and Part-of-Speech tagging.

Table 2 – Preprocessing techniques used in selected articles.

Preprocessing Technique	**No. of Articles**
Stop words removal	8
Special characters removal	6
Stemming	5
Tokenization	5
Data Auto Correction	4
Lowerization	4
Lemmatization	3
Part-of-Speech tagging	3

Regarding feature extraction techniques, conclusions can be drawn through Table 3. Guru *et al*., (2018) proposed a feature extraction method where a Bag of Words (BoW) is created, after the preprocessing, to designate each class. The advantages of using this approach rely on the importance of isolating the terms that characterize the complaints of each class. Thus, Term Class Weight-Inverse Class Frequency (TCW-ICF) comprises two different calculations: Term Class Weight (TCW) consists of dividing the frequency of a term in a class by the frequency of all terms in that specific class; and the Inverse Class Frequency (ICF) that assigns more importance to the term if it is present in only one class than if it is in all classes. The terms with a high value of TCW-ICF are highly likely to discriminate the class.

More widely used is the Term Frequency-Inverse Document Frequency (TF-IDF), and unlike the technique presented before, this measures the importance of a term based on the number of times it appears in the text, but it is conditioned by the number of documents it appears in, meaning a word present in many complaints is less relevant than a word in few

(Peng *et al.*, 2022). The second most mentioned technique is BoW, which calculates the relative frequency of a term per document (Goncarovs, 2019).

Table 3 - Feature extraction techniques used in selected articles.

Feature Extraction Technique	No. of Articles
TF-IDF	11
Bag of Words	5
Word2vec	2
BERT	1
Count Vectorizer	1
TCW-ICF	1

Several modelling approaches found in the reviewed articles are enumerated in Table 4, and it is notorious for the diversity of models used in the task of Text Classification, from classical ML algorithms to Transfer Learning ones. The most widely used model was the Support Vector Machine, followed by BERT. On the other hand, almost half of the models were only covered in a single article.

Table 4 - Modelling techniques used in selected articles.

Machine Learning Algorithm	No. of Articles	Machine Learning Algorithm	No. of Articles
Support Vector Machine	10	TextCNN	2
Bidirectional Encoder Representations from Transformers	7	Artificial Neural Networks	1
Convolutional Neural Network	7	Backpropagation Multilabel Learning	1
Long Short-Term Memory	6	Bayes Network	1
Multinomial Naïve Bayes	6	Chain Classifier	1
k-Nearest Neighbors	5	Gaussian Naïve Bayes	1
Random Forest	5	LightGBM	1
Logistic Regression	4	RAndom k labELsets	1
Sequential Minimal Optimization	4	SimpleLogistic	1
Decision Tree	2	Tree Augmented Naïve Bayes Transfer Classification	1
Gated Recurrent Unit	2	XGBoost	1

Guru *et al*., (2018) conducted an experiment with SVM and k-Nearest Neighbor (kNN) algorithms to classify farmers' complaints written in Arabic language. The complaints have gone through preprocessing consisting of removing stop words, correcting misspelled words and stemming. For feature extraction, two different techniques were used: TF-IDF with the optimal set of features being selected through Information Gain (IG), Bi-Normal Separation (BNS), Chi Square (CHI) and Weighted Log Likelihood Ratio (WLLR); and TCW-ICF, as

mentioned before. The results of the TF-IDF experiments vary according to the number of features selected and the method used. The best result was achieved by CHI with an Accuracy of 84.62% and the one with the lowest performance was BNS with 81.60%. However, the experiment with TCW-ICF reached the highest Accuracy, 85.08%. These results were obtained with kNN since it outperformed SVM.

HaCohen-Kerner *et al*., (2019) proposed the automatic classification of complaint letters written in Hebrew by recurring to four ML algorithms: Bayes Network (BN), Random Forest, Sequential Minimal Optimization (SMO), and SimpleLogistic (SL). The set of experiments started with seven classes and gradually decreased to four by eliminating the category with the highest misclassification. By doing so, the Accuracy rose from 84.5% to 93.8%. The features were selected through BoW, considering multiple sets of word unigrams, and the presence of stop words was tested to determine its influence on the performance of the models. The division in train and test sets was 67% to 33%, respectively, and the best result in all sets was obtained by SimpleLogistic in the experiments without stop words.

When it comes to Transfer Learning approaches, Matos *et al*., (2022) experimented BERT pre-trained models, more specifically the BERT-LinearLayer, the BERT-CNN, and the GAN-BERT, to identify online hate speech in Youtube comments written in Portuguese. These comments were manually classified according to the existence of hate speech and its type. The preprocessing consisted of anonymizing the users mentioned in the comments and removing repeated punctuation and emojis. An experiment to perceive the influence of preprocessing was carried out, and the GAN-BERT model was the only one to underperform. The best performing model was the BERT-CNN with an F1-Score of 0.721. Although this article is not directly related to complaints, considering the result of the above queries, it is the only one that approaches the classification of text written in Portuguese. Therefore, this article was considered for analysis.

CHAPTER 2

Data Preparation

The original dataset contains data concerning the 47025 complaints addressed to Worten in 2020 and 2021. It consists of 18 attributes, including the complaints' case number, the store where it occurred (including physical stores in mainland Spain, the Canary Islands and Portugal, and online stores in the three regions), and the date of opening and closing of the complaint resolution process. Customers can make complaints about Worten's services through social media, *Associação Portuguesa para a Defesa do Consumidor* (DECO), the largest Portuguese consumer protection organization, the complaints book, or verbally, as stated in the complaint's reception channel attribute, among other options. Table 5 lists all the attributes with a brief explanation and a sample of the values.

Table 5 - Identification of the 18 attributes of the dataset.

Attribute	Description	Value
Case number	Distinct identification value	1766040
Store code	Numeric code of the store	1460
Store description	Description of the store's name	Worten Online PT
Date/ Time of opening	Date and time of opening of the process	4/5/20 10:15
Date/ Time of closing	Date and time of closing of the process	19/5/20 13:42
Created by	Name of the entity that initiated the process	System
Reception channel	Reception channel of the complaint process	Livro de Reclamações Eletrónico
Type	Categorization of the type of complaint	Cartões e Meios de Pagamento
Reason	Sub-categorization of the type of complaint	Pagamento por multibanco indisponível
Area/ Detail	Specification of where the problem occurred	Multibanco
Business Unit	Description of the product's Business Unit	Grandes Domésticos
Category	Description of the product's Category	Frio
Status	Identification of the complaint process status	Arquivado
Brand	Name of the brand of the product	LG
SPV Process	After-sales service case number	NaN
Order number	Number of the order placed by the client	32945403
Delivery process no.	Number of the process delivery	NaN
Description	Description of the complaint	No dia 1 de maio fiz a compra de um frigorífico (…)

It also specifies the type of complaint, whether it is related to the product's quality, satisfaction and return policy, or payment methods, for instance. The attribute *Reason* is more specific since it subdivides the *Type* attribute into different classes. Monthly fees, commissions,

and promotional code validity are all examples of complaints' reasons regarding the "Payment methods" type of complaint. The *Area/ Detail* column describes if the complaint concerns the Automated Teller Machine (ATM), a promotional code, or a cashier's check, as exemplified.

The remaining attributes include the Business Unit and corresponding Category to which the product or service being complained about belongs; the status of the complaint analysis process, whether it is in progress, has been archived, reopened, or closed; the brand of the product and the order number. The most important attribute is the description of the complaint since it is the variable to be classified.

In this dataset, there are three types of complaints: the clients write some, others are a narration of the complaint that the customer made by another means, and a few contain just attachments (Table 6).

Table 6 - Difference between the existing types of complaints.

Type of Complaint	Value
Real complaint	Venho por este meio informar o meu descontentamento (…)
Narration of complaint	Cliente reclama o facto de ter feito uma compra (…)
Containing only annexes	ANEXO

Considering that the person who transcribes the complaints by telephone or verbally, immediately identifies the complaints' topic, it is not meaningful to categorize these complaints. Therefore, the complaints were filtered through the following regular expression: ^[Cc]liente[s]* and the ones starting with the word "Cliente" in all forms, were excluded.

There are also complaints that the text only consists of the word "ANEXO". Since this word could apply to all topics, it does not add value, and these complaints were also eliminated. Then, a new dataset, with 29968 records, was created, excluding the ones considered in the previous examples.

Since complaints referring to stores in Spain are most likely written in Spanish, *langid.py*, a tool that identifies the language in which a document is written, was used. The results confirmed that there were indeed complaints in Spanish but also in English and French, indicating that they may have been made by foreigners living or visiting these two countries. These complaints were eliminated, as they represented merely about 6.32% of the total complaints. The final dataset consists of 28074 records.

In order to clean up all the noise in the complaints, preprocessing was applied. First, words beginning with "https" or "www" without a blank space following, indicating a hyperlink, were

eliminated, as well as punctuation and any digits present in the complaints. The stop words were also excluded to help reduce the dimensionality of the features going into the models (Guru *et al.*, 2018), as they added no value.

Subsequently, *Stanza*[1], a Natural Language Processing tool for tokenizing and lemmatizing texts in Portuguese, among other languages (Freitas & Souza, 2023), was tested in the complaints. It divides the sentences into tokens and then returns the canonical form of words as part of the lemmatization process (Qi *et al*., 2020). The uppercase letters found in the complaints were also transformed into lowercase (HaCohen-Kernera *et al*., 2019) to homogenize the features.

Given the size of the dataset, the number of columns was reduced to eight. The ones that prevailed were the *Store* of occurrence and the *Date* the complaint process was opened, the *Type* and *Reason* of the complaint, the *Business Unit*, the *Category* and *Brand* to which it refers, and the complaint *Description*.

The number of missing values in each column was determined, where the most impacted variables were the *Business Unit*, *Category,* and *Brand*, with 3440 being the highest number of nulls. In order not to lose records and given the irrelevance of the data in these variables, the nulls were replaced by "undefined". The other columns did not undergo any data transformation except for the *Date* column, where the time of occurrence of the complaint was removed, to reduce the heterogeneity of values, and only the date remained.

[1] See Vajjala S., Majumder B. & Gupta A. (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly.

CHAPTER 3

Data Understanding

The Data Understanding stage was essential to perceive how the data behaves. In the first step, the ten thousand most relevant keywords were extracted through the TF-IDF and then represented in clusters. The Elbow rule was employed to support the optimal number of clusters, and three was found to be the optimal number (the first kink in the line plot - Figure 1).

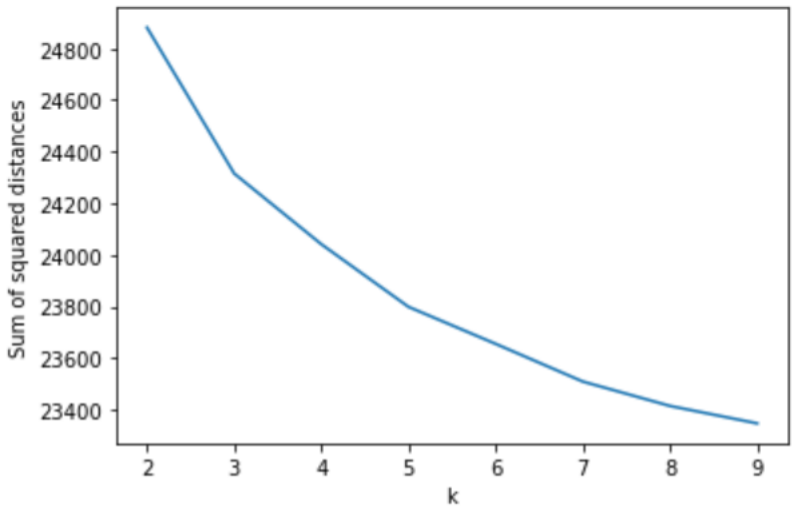

Figure 1 - Identification of the number of clusters based on the "Elbow Method".

By analysing the results shown in Figure 2, it is noticeable that it is not possible to distinguish each cluster according to one topic nor label them. In fact, there are several repeated words between the three clusters, the most frequent being "cliente", "reclamação", "Worten" and "anexar". This underlines the fact that the complaints are very diverse from one another and that customers do not write in the same manner when complaining about the same topic.

Figure 2 - Clusters generated from the 10,000 most relevant features.

Since the previous analysis did not allow many conclusions to be drawn, a different approach was adopted. For all columns of the dataset except the column containing the complaint description, the number of unique values was counted, and generated a cluster for each value with more than five hundred complaints. As there is no value in the *Date* column with more than five hundred complaints, only dates with more than one hundred complaints were considered. When looking at the results regarding the *Store* column in Figure 3, it is noticeable that in the online store clusters, the word "marketplace", Worten's digital platform in which selected companies sell their articles, stands out, as well as "fornecedor", "fatura", "preço" and "previsão". These might be indicators of dissatisfaction with the delivery time and price of the items.

Figure 3 - Clusters generated from the *Store* attribute.

At the company headquarters' cluster, the words are more bureaucratic, with "cartão", "instalação", "segurar" e "contratar" being among the most recurrent words. Regarding the complaints made at the WRT Lisboa store, the words differentiating this cluster are "Apple", "stock", "colaborador", since this word is only applicable to complaints about physical stores, and "avariar". Nevertheless, it is not yet possible to clearly define these clusters solely based on these words.

The column *Type* originated eight clusters and the words stand out more than in the other column clusters. This implies that these words were used more often to describe a complaint of the same type, which can help classify the complaint. By looking at Figure 4, in the cluster "Atendimento", "reclamação" was the most written word, while in the "Instalação de Artigos," the word "técnico" highlights, as could be expected. Although specific keywords exist in each

cluster, a few are repeated, such as "máquina" and "telemóvel" making it difficult to identify each cluster. As for the clusters of the other columns, the results are shown in Figures A1-A5 of the Appendix, but like the previous examples, it is not very straightforward to label them.

Figure 4 - Clusters generated by the attribute *Type*.

In addition to the previous experiment, another set of clusters was formed, but this time, considering the number of words of each complaint. The results (Figure 5) show that the cluster for complaints with less than ten words concerns mainly documentation, attachments, and scans. While in the other clusters, the disparity of words increases with the number of words the complaint has, emphasizing that complaints with more words become more complex. In this dataset, 12744 complaints are between twenty to one hundred words long, most of the records, followed by complaints with more than one hundred words. The minority of the complaints, 1372, are written with ten to twenty words.

Figure 5 - Clusters generated according to complaints' number of words.

An exploratory data analysis was then conducted, and the five predominant values of each attribute were plotted in a bar chart, to perceive their distribution. For the *Store* column, the two most dominant expressions were "Worten Online PT" followed by "WRT ON Line", as both refer to the online store, were joined in one column (Figure 6). It is noticeable the difference between the number of complaints in the online store and the four physical stores with more complaints, as it surpasses the number of complaints in WRT Lisboa, WRT Leiria, WRT Fórum Montijo and WRT Seixal combined. The most prominent *Reason* of discontent is due to failed or faulty delivery, and the product category with the most complaints is "Desbloqueados", which includes all phones with no associated carrier. In what comes to the *Brand* the most affected is "Samsung", followed by "Apple" and "LG", all phone brands. These previous results can be observed in the Appendix (Figures B1-B5).

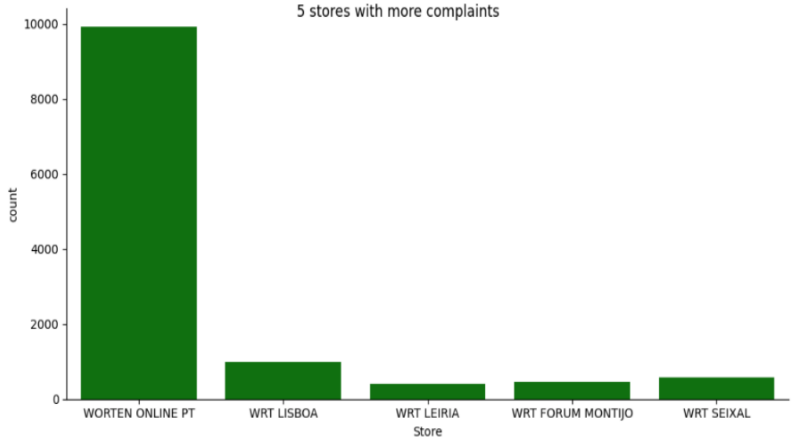

Figure 6 - Distribution of top five stores with more complaints.

Regarding the *Date* column, it was plotted the evolution of the number of complaints throughout time, visible in Figure 7. The highest number of complaints was made on January 26, 2021, with 179 complaints. This can be explained by the fact that it is the month after Christmas, a period known for significant sales, giving people time to try out products and evaluate if there are any faults. The other complaint peaks can be observed during November and December, the post Black Friday time, and in mid-March when the quarantine due to Covid-19 started.

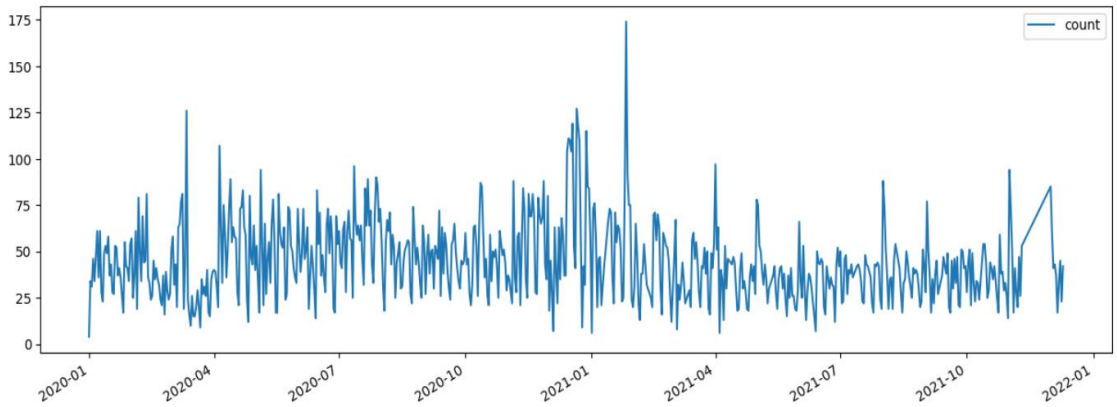

Figure 7 - Evolution of the number of complaints between 2020 and 2021

Since this study aims to automatically classify the complaints according to their topic, the classes had to be defined. The attribute *Type* is broader and has 24 unique values, whereas the attribute *Reason*, with 118 unique values, is a subcategorization of the previous column, which means the values are more specific. Thus, to transform the attribute *Reason* into smaller classes, it was necessary to manually categorize all the values and fit them into one single class. After this task, six classes resulted: *Money*, *Time*, *Service*, *Client*, *Technical Problem* and *Other*, this latter class is vaguer and broader since it comprises the complaints' reasons that did not fit the other classes. The detailed division of the attribute *Reason* into classes can be found in the Appendix (Tables C1-C6). Given that the number of complaints in the six classes is different, it can be inferred that the classes are imbalanced. The *Service* class has the most complaints, 8631, while the *Money* class has the least with just 2646 complaints.

In order to detect the most dominant words in each class, pie charts were plotted to represent them. Considering the results, most of the ten most recurrent words are of low relevance. However, it is possible to draw some conclusions. The word "equipamento" only figures in the *Technical Problem* class and the word "dia" is more common in the *Time* class than in the other classes (Figure 8). In the remaining, no differentiating words can be found. The additional Figures are in the Appendix (Figures D1-D5).

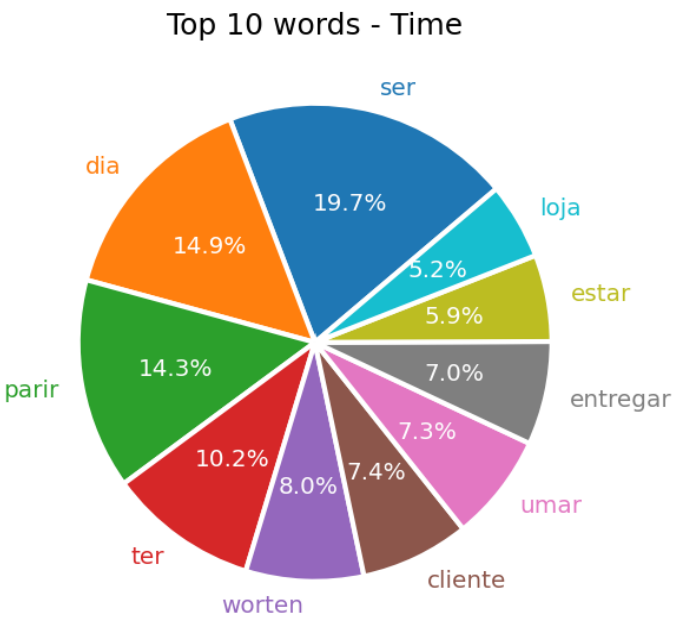

Figure 8 - Distribution of the top ten words in the *Time* class.

CHAPTER 4

Modelling

In this Chapter, all the ML algorithms used to classify complaints according to their class are presented, namely, Decision Tree, Support Vector Machine, Logistic Regression, Random Forest, BERT[2], Multinomial Naïve Bayes[3], XGBoost, LightGBM[4] and CatBoost[5]. All the experiments performed are described in detail, whether with TF-IDF or Bag of Words to extract the keywords, using balanced or imbalanced data, as well as experiments with pre-trained models.

The first classifier to be experimented with was the Decision Tree algorithm, which can handle atypical values (McArthur *et al*., 2018) and receives either numerical or text data as input. The tree data structure mainly consists of nodes where the features are evaluated according to the rules defined in the branches and sorted into the corresponding category, the leaf nodes (Goncarovs, 2019).

Then, the Support Vector Machine algorithm was tested. This classifier performs well on data with high dimensionality (Fatima & Srinivasu, 2017) and creates a surface that increases the distance between the classes. The support vectors define this surface, also known as hyperplane (Cervantes *et al*., 2020).

The Logistic Regression algorithm has been used extensively for some years and, unlike the previous algorithms, is based on statistical methods (Shah, *et al*., 2020). The prediction of each class, in the shape of a vector of words, is made considering the weight of each variable in the set of variables (Bangyal *et al*., 2021).

Concerning the Multinomial Naïve Bayes, this algorithm considers the number of times the word appears in a document, that is, the term frequency. It identifies the occurrence or non-occurrence of the word in the document and its corresponding frequency (Singh *et al*.,

[2] See Géron A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly.
[3] See Vajjala S., Majumder B. & Gupta A. (2020). Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. O'Reilly.
[4] See Quinto B. (2020). Next-Generation Machine Learning with Spark: Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More. O'Reilly.
[5] See George N. (2021). Practical Data Science with Python. O'Reilly.

2019), which is context-independent. The non-linkage between the size of the document and its classes is another premise of this classifier (Pratama & Purwarianti, 2017).

The Random Forest classifier comprises multiple trees built from the Decision Tree algorithm (HaCohen-Kernera et al., 2019) and can cope with many input variables. These trees are independent from each other, implying that a tree can be stronger than the others given the error rate, and the correlation between them can affect the performance of the classifier (Shah, *et al*., 2020).

Another classifier used was the XGBoost, also known as Extreme Gradient Boosting, which aggregates several decision trees generated from the gradient descent principle, with poor accuracy results, into a more accurate model. The loss error becomes smaller as the decision trees are iterated through (Qi, 2020).

As an upgrade of the previous classifier, LightGBM is composed of various decision trees but performs better as it is more robust (Tang *et al*., 2020). It is intended to solve the dimensionality problems and improve how node partitions are done (Gu *et al*., 2023).

The CatBoost algorithm has a much better prediction time compared with the two-gradient boosting-based models, previously mentioned, since it uses symmetric trees. These trees are built sequentially through pre-defined parameters (Nandy & Kumar, 2021) and are not likely to be overfitted. This classifier performs better than other ML algorithms in multi-class classification tasks involving imbalanced data (Aldania *et al*., 2021).

Lastly, the neural network BERT is intended to handle various NLP-related tasks (Khadhraoui *et al*., 2022). This model consists of a pre-training phase, in which bidirectional representations are trained on unlabeled data, and a fine-tuning phase using labeled data where the parameters are tuned (González-Carvajal & Garrido-Merchán, 2021). The results of this model are quite positive, considering a variety of ML processes (Qasim *et al*., 2022).

In order to be able to compare the results of the different trials somewhat with each other, the data was divided into a train and a test set in the proportion 70% to 30%, in all the experiments. As a multi-class classification problem involving categorical variables, one-hot encoding was used to transform them into a numeric array. In Figure 9, it is possible to observe the flowchart of all sixteen experiments performed.

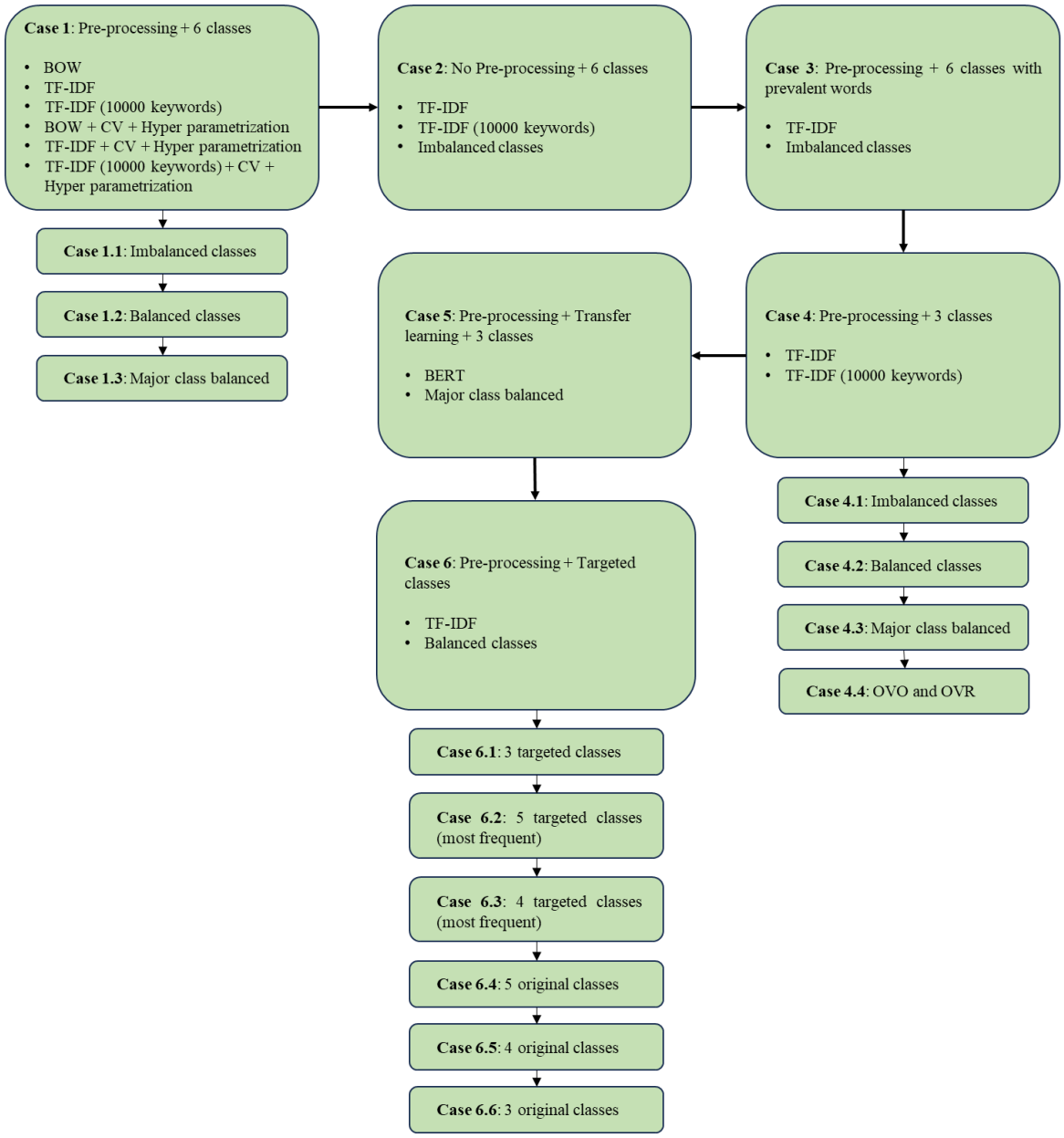

Figure 9 - Flowchart of all the Cases experimented.

4.1. Case 1: Preprocessing + 6 classes

The first Case to be conducted involved the classification of six classes, and the keywords were extracted through Bag of Words and TF-IDF. It also included limiting the most relevant keywords, cross-validation (CV), and hyperparametrization. In this Case, three different experiments were performed, one with imbalanced classes, another with balanced classes considering the number of records in the minority class, and the last one, with the majority class balanced.

4.1.1. Case 1.1: Preprocessing + 6 imbalanced classes

Firstly, the keywords were selected using Bag of Words, and TF-IDF without limitation since they were all input to the ML models, and then only the ten thousand keywords with the highest TF-IDF score were used. Afterward, these three experiments were repeated, tuning the hyperparameters through grid search, using a five-fold cross-validation. This Case was carried out with the six original imbalanced classes.

4.1.2. Case 1.2: Preprocessing + 6 balanced classes

The six experiments mentioned above were rerun, with all classes equally balanced to the class with the fewest records, *Money*, with only 2646. As such, the dataset size has reduced to 15876 complaints, considering that the classes got the same number of records. The class that suffered the most significant decrease was *Service,* which had 8631 records, more than the two minority classes combined, and lost almost 6000 complaints. The balancing was done using *RandomUnderSampler*[6].

4.1.3. Case 1.3: Preprocessing + 6 classes (major class balanced)

As the previous dataset transformation resulted in a significant loss of records, another Case was tested but only with the predominant class balanced, following the preceding method. That is, only the class with the highest number of records was balanced to equal the number of records in the minority class. This balancing will be referred to as major class balanced.

Thus, *Service* has come to have 2646 complaints, the number of records in the minority class, and the others remained the same. The dominant class became the *Technical Problem* class with 5245 records, resulting in the dataset having 22089 complaints. The six previous experiments were applied to this new dataset.

[6] See https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html

4.2. Case 2: No Preprocessing + 6 classes

In order to perceive if the preprocessing stage influences the performance of the models, an experiment was tested without preprocessed data. For this Case, only two trials were conducted: with and without limitation of the keywords to the ten thousand most relevant ones, using TF-IDF and the dataset with the imbalanced classes.

4.3. Case 3: Preprocessing + 6 classes with prevalent words

In this Case, all complaints were reduced to the most prevalent words in each class. To do so, the hundred words that appeared most frequently in each class were listed and manually selected for each class. Words like "Worten", "cliente", "loja" are common to all classes and, hence, were immediately excluded since they are not able to differentiate each class. The keywords assigned to each class can be found in Table 7.

Table 7 - Definition of keywords for each class.

Class	Keywords
Service	Entrega, entregue, online, serviço, transportadora, vendedor
Technical Problem	Equipamento, reparação, garantia, avaria, stock, técnico
Time	Dia, hoje, data, úteis, instalação, tempo, aguardar, espera, prazo
Client	Atendimento, informação, prioritário
Money	Valor, reembolso, devolução, apoio, dinheiro, preço, pagamento, cartão, recebido
Other	Troca, clientes, comportamento, publicidade, instruções

The words chosen to limit the text of the *Service* class complaints are all related to product delivery services, while the class *Technical Problem* has more to do with repairs and problems that occur, and the company oversees solving them. The class *Client* has less keywords than the remaining because the top hundred words were coincident with the most common words of the others. The keywords from the *Other* class could apply to the rest of the classes, as they are more diverse. The dataset used in this Case had imbalanced classes and preprocessing applied, and TF-IDF was employed to extract the keywords.

4.4. Case 4: Preprocessing + 3 classes

For this Case, the number of classes was reduced to three. Therefore, the class *Client* now includes the classes *Time* and *Money*, the class *Technical Problem* absorbed the *Other* class, and the class *Service* remained the same since it is the one that has the highest number of records. This division considered the content of each class and the number of records, so they were as balanced as possible.

Other experiments ran with imbalanced and balanced data, and only with the majority class balanced. An additional experiment with equally balanced data, One-vs-One (OVO), and One-vs-Rest (OVR) methods was also tested.

4.4.1. Case 4.1: Preprocessing + 3 imbalanced classes

The first experiment involved a dataset with slightly imbalanced classes in which the *Client/ Time/ Money* class had 10619 records, and the *Technical Problem/ Other* and *Service* classes had around 8,000 each. The input keywords for the models were selected with TF-IDF, initially without limit and then considering the ten thousand most important ones.

4.4.2. Case 4.2: Preprocessing + 3 balanced classes

By equally balancing the classes according to the class with the lowest number of complaints, the dataset came to have 25893 complaints, only about two thousand fewer records than the dataset used in the previous experiment. The same methods were employed for this second Case to select the features.

4.4.3. Case 4.3: Preprocessing + 3 balanced classes (major class balanced)

Similar to the experiments of Case 1, a new dataset was created by balancing the class with the highest number of complaints. Hence, the number of records in the *Client/ Time/ Money* class was reduced to 8631 to equal the *Service* class. In this dataset, the classes are nearly equally balanced since the class *Technical Problem/ Other* has less than two hundred more complaints than the remaining classes.

4.4.4. Case 4.4: Preprocessing + 3 balanced classes + OVO and OVR

In Case 4 an additional experiment was performed with the classes equally balanced, OVO and OVR methods, to transform the multi-class classification into a binary classification problem. While in the first approach, one class is considered positive and the other negative, in the second approach, a class is treated as positive and the remaining classes are treated as negative (Jia *et al*., 2023).

4.5. Case 5: Transfer learning + 3 classes

Following the experiment with three classes, a pre-trained BERT model was tested. The classifier chosen for this purpose was pedro-m4u/1000_respostas-MODELO_2[7]. This classifier was tested on a multi-class classification problem and trained on a corpus of sentences written in Portuguese, achieving an Accuracy of 86.3%. The dataset used in the experiment only had the majority class balanced.

4.6. Case 6: Preprocessing + targeted classes

As for the last Case, the classes were redefined once again to understand the models' behavior when the classes were more particular and differentiated. Thus, various experiments were tested with three to five targeted classes and then with the original classes.

4.6.1. Case 6.1: Preprocessing + 3 targeted classes

The attribute *Reason* was revised, and three values from distinct classes were selected: "Política de Preço Mínimo Garantido" from *Money* class, "Rutura de Stock" from *Technical Problem class* and "Demora no Atendimento" from the *Time* class. Since the value "Política de Preço Mínimo Garantido" had fewer records than the others, the dataset was balanced given the minority class, making a total of about six hundred records.

[7] See https://huggingface.co/pedro-m4u/1000_respostas-MODELO_2

4.6.2. Case 6.2: Preprocessing + 5 targeted major classes

Since the previous dataset had a very limited number of records, a new one was created with the values that had the most complaints from each class: “Artigo fora da política de satisfação/ devolução” from *Other* class, “Avaria Reincidente” from *Technical Problem* class, “Entrega não realizada/ com problemas” from *Service* class, “Informações Prestadas” from *Client* class and “Demora no reembolso” from *Time* class.

The value "Política de Preço Mínimo Garantido", despite being the one with more complaints in the *Money* class, only had about two hundred complaints and would be responsible for reducing the number of records in the dataset after balancing it based on the minority class. For this reason, it was excluded. After balancing the dataset, the number of records was 4905, much larger than the previous experiment.

4.6.3. Case 6.3: Preprocessing + 4 targeted classes

Another experiment was tested using the previous targeted classes, but without "Informações prestadas" which belongs to the *Client* class. This one is very subjective since the provided information could be about topics regarding the other classes, so it was removed. With this transformation, the number of complaints in the dataset has decreased to 3924.

4.6.4. Case 6.4: Preprocessing + 5 original classes

Following the strategy above, a different experiment was performed with the original classes, excluding the *Client* one for being more diverse. Thus, by balancing all the classes, the dataset came to have 4060 complaints, and the classes to be classified were *Service*, *Money*, *Technical Problem*, *Other,* and *Time*.

4.6.5. Case 6.5: Preprocessing + 4 original classes

The class *Other* was eliminated to determine if removing classes with more diverse topics would positively influence the results. This class comprises values related to air quality, lack of resolution, and when the customer does not describe the reason for dissatisfaction. There are many different topics, so it becomes difficult to accurately classify this class.

4.6.6. Case 6.6: Preprocessing + 3 original classes

Lastly, for the last experiment, the *Service* class was excluded, due to its topics including quality of service and incorrectly performed service. The remaining ones were *Money*, *Time* and *Technical Problem,* which formed a balanced dataset with 2436 complaints. No further experiments were carried out since there was no meaning in decomposing these three classes.

CHAPTER 5

Results and Discussion

This Chapter describes all the results obtained in various experiments, with or without preprocessing and using TF-IDF or Bag of Word to extract the features. The selection of hyperparameters and the results of the experiments with a different number of classes are presented.

5.1. Case 1: Preprocessing + 6 classes

5.1.1. Case 1.1: Preprocessing + 6 imbalanced classes

Since the classes in the dataset used in Case 1.1 were not balanced, Precision was the considered metric to evaluate the performance of the models. Regarding the first experiments involving BoW and TF-IDF to extract the keywords, it is noticeable that TF-IDF performed better than BoW in most cases, and when this occurred, the difference was quite significant, exceeding a maximum of 11 percentage points with the Multinomial Naïve Bayes classifier (Table 8). On the other hand, when BoW performed better, the results were very close to those achieved by TF-IDF.

The Multinomial Naïve Bayes classifier reached the best Precision with 53.92%, and Catboost was the only one to get to 50%. The Decision Tree classifier had just over 30%, the lowest performance. Half of the models performed slightly better in the third experiment with TF-IDF and a limitation of ten thousand keywords. Still, the best improvement in the Random Forest classifier was not even 2 percentage points.

Regarding the experiments with hyperparametrization and cross-validation, the results were superior to the previous ones in all experiments, except for the experiment with BoW and the Multinomial Naïve Bayes classifier, and the experiment with TF-IDF limited by ten thousand keywords and the Logistic Regression classifier, in which the hyperparameters found through grid-search were the default ones. Therefore, the results were the same as in the experiments without hyperparametrization (see Table 8).

Unlike the other experiments with the different classifiers, the Multinomial Naïve Bayes decreased its Precision by around 9 percentual points in the experiment with TF-IDF using hyperparameters, compared to the experiment with the same feature extraction technique and without hyperparameterization. Since the Accuracy, Recall, and F1-Measure values achieved by this classifier in the experiment with TF-IDF were relatively low, reaching around 13% of F1-Measure, the Precision of 53.9% will be excluded as a positive result.

The Decision Tree classifier, which had the worst result before, achieved a Precision of 57.75%, the third greatest, while the Random Forest classifier had the best performance with the TF-IDF experiments: 62.23% without limitation and 75.54% with ten thousand keywords. Although these results were the highest, they can not be considered positive due to the lower Accuracy, Recall and F1-measure results. Hence, the best result, 50.46%, was achieved by the Support Vector Machine classifier in the experiment with TF-IDF, hyperparametrization (*max_iter* = 100, *C* = 0.1, *intercept_scaling* = 1, *loss* = "squared_hinge", *multi_class* = "ovr", *penalty* = "l2", *tol* = 0.0001) and five-fold cross-validation. Concerning the other classifiers, Precision was close to 50%, meaning that only half of the complaints were classified correctly.

The last three experiments for each of the cases of Case 1 with XGBoost, LightGBM and Catboost, powerful gradient boosting methods, were only performed after running Case 4, in an attempt to obtain better results. As the Precision values did not stand out compared to the experiments with the initial algorithms, experiments with cross-validation and hyperparameterization were not conducted.

Table 8 - Results of Case 1.1 (preprocessing + 6 imbalanced classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + BoW	Decision Tree	31.60%	32.51%	29.34%	30.79%
Preprocessing + TF-IDF		33.67%	31.16%	30.60%	30.77%
Preprocessing + TF-IDF (10000 keywords)		34.22%	31.57%	30.88%	31.12%
Preprocessing + BoW + CV + Hyperparametrization		12.88%	38.94%	9.35%	13.86%
Preprocessing + TF-IDF + CV + Hyperparametrization		36.73%	35.24%	29.00%	26.25%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		35.94%	57.75%	25.68%	21.67%
Preprocessing + BoW	Support Vector Machine	39.10%	37.82%	36.50%	37.01%
Preprocessing + TF-IDF		46.23%	45.43%	45.43%	43.36%
Preprocessing + TF-IDF (10000 keywords)		45.77%	44.89%	42.11%	42.96%
Preprocessing + BoW + CV + Hyperparametrization		43.22%	42.13%	40.10%	40.78%
Preprocessing + TF-IDF + CV + Hyperparametrization		49.64%	50.46%	44.29%	45.44%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		49.74%	50.45%	44.46%	45.60%
Preprocessing + BoW	Logistic Regression	43.35%	42.58%	40.38%	41.21%
Preprocessing + TF-IDF		49.20%	49.75%	44.20%	45.67%
Preprocessing + TF-IDF (10000 keywords)		49.12%	49.64%	44.17%	45.64%
Preprocessing + BoW + CV + Hyperparametrization		47.64%	49.56%	41.58%	43.26%
Preprocessing + TF-IDF + CV + Hyperparametrization		49.21%	49.77%	44.21%	45.67%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		49.12%	49.64%	44.17%	45.64%
Preprocessing +BoW	Multinomial Naïve Bayes	43.20%	42.52%	42.07%	41.72%
Preprocessing + TF-IDF		33.60%	53.92%	19.33%	12.78%
Preprocessing + TF-IDF (10000 keywords)		37.72%	48.36%	24.97%	21.73%
Preprocessing + BoW + CV + Hyperparametrization		43.20%	42.52%	42.07%	41.72%
Preprocessing + TF-IDF + CV + Hyperparametrization		42.32%	45.04%	38.11%	37.28%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		37.72%	48.36%	24.97%	21.73%
Preprocessing +BoW	Random Forest	42.29%	48.40%	31.81%	32.07%
Preprocessing + TF-IDF		41.36%	47.84%	30.49%	30.55%

Preprocessing + TF-IDF (10000 keywords)		42.92%	49.52%	32.78%	33.27%
Preprocessing + BoW + CV + Hyperparametrization		36.16%	62.23%	22.50%	17.69%
Preprocessing + TF-IDF + CV + Hyperparametrization		35.46%	75.54%	21.39%	15.82%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		37.59%	61.35%	24.24%	20.21%
Preprocessing + BoW	XGBoost	47.90%	48.30%	42.25%	43.41%
Preprocessing + TF-IDF		47.68%	48.10%	42.20%	43.53%
Preprocessing + TF-IDF (10000 keywords)		47.79%	48.25%	42.39%	43.74%
Preprocessing +BoW	LightGBM	48.96%	49.57%	43.52%	44.88%
Preprocessing + TF-IDF		49.40%	49.81%	44.25%	45.61%
Preprocessing + TF-IDF (10000 keywords)		48.96%	49.11%	43.76%	44.99%
Preprocessing + BoW	CatBoost	48.17%	49.63%	42.19%	43.67%
Preprocessing + TF-IDF		49.27%	50.10%	43.62%	45.07%
Preprocessing + TF-IDF (10000 keywords)		49.17%	50.12%	43.56%	45.00%

5.1.2. Case 1.2: Preprocessing + 6 balanced classes

The following experiment to be performed was with a balanced dataset, and Accuracy was considered the most appropriate metric to evaluate the performance of the classifiers. Similarly to the last experiment, Bag of Words was applied to extract the features, and the best result did not reach 46% of Accuracy, achieved by the LightGBM classifier (Table 9). In what comes to the worst results, the Decision Tree, as before, had the lowest performance with 29.48% and was the only classifier to score below 30% of Accuracy.

By applying the TF-IDF, the results generally improved except for the Multinomial Naïve Bayes, where the results dropped by 2 percentage points, and the XGBoost, which came very close to the Bag of Words experiment results. On the other hand, the Support Vector Machine achieved 44.05% of Accuracy, representing an increase of almost 7 percentage points, the most significant rise.

Another experiment was run to determine if the number of features in the model would impact its performance. By analysing the Accuracy values, the results in the experiments with a limitation on the number of keywords were worse than without restriction in all experiments except for the Decision Tree and the Catboost classifiers. However, the difference in the results between the two experiments was residual, as it was less than 1 percentage point.

Regarding the hyperparametrization experiments, the results improved in only two classifiers: Decision Tree, which has increased by 8 percentage points compared to the experiment with TF-IDF and ten thousand keywords, without hyperparameters, and the Support Vector Machine classifier that improved its performance in all experiments achieving an Accuracy of 46.17%. As for the Logistic Regression, the results with hyperparameters only enhanced in the BoW experiment, and the Multinomial Naïve Bayes and Random Forest classifiers performed worse in all experiments.

Hence, in this Case, the best result was achieved by the Logistic Regression algorithm in the experiment with TF-IDF and no limitation of the number of keywords with an Accuracy of 46.53%. Unlike the previous Case, there were no experiments with high Precision values and low results in the other metrics.

Table 9 - Results of Case 1.2 (preprocessing + 6 balanced classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + BoW	Decision Tree	29.48%	32.68%	29.45%	30.95%
Preprocessing + TF-IDF		31.07%	30.92%	31.04%	30.88%
Preprocessing + TF-IDF (10000 keywords)		31.30%	31.56%	31.29%	31.31%
Preprocessing + BoW + CV + Hyperparametrization		29.90%	32.90%	29.90%	31.46%
Preprocessing + TF-IDF + CV + Hyperparametrization		31.95%	31.31%	31.94%	28.86%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		39.30%	38.49%	39.39%	37.14%
Preprocessing + BoW	Support Vector Machine	37.20%	37.33%	37.23%	37.18%
Preprocessing + TF-IDF		44.05%	43.78%	44.07%	43.79%
Preprocessing + TF-IDF (10000 keywords)		43.84%	43.68%	43.86%	43.86%
Preprocessing + BoW + CV + Hyperparametrization		41.21%	41.20%	41.23%	41.07%
Preprocessing + TF-IDF + CV + Hyperparametrization		46.00%	45.32%	46.01%	45.23%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		46.17%	45.54%	46.18%	45.44%
Preprocessing + BoW	Logistic Regression	42.10%	42.34%	42.12%	42.08%
Preprocessing + TF-IDF		46.53%	46.28%	46.56%	46.24%
Preprocessing + TF-IDF (10000 keywords)		46.38%	46.35%	46.37%	46.18%
Preprocessing + BoW + CV + Hyperparametrization		45.31%	46.28%	45.31%	45.04%
Preprocessing + TF-IDF + CV + Hyperparametrization		46.29%	46.32%	46.29%	46.27%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		46.38%	46.35%	46.37%	46.18%
Preprocessing +BoW	Multinomial Naïve Bayes	42.75%	42.33%	42.87%	41.84%
Preprocessing + TF-IDF		41.34%	40.70%	41.50%	39.01%
Preprocessing + TF-IDF (10000 keywords)		40.69%	39.39%	40.86%	38.48%
Preprocessing + BoW + CV + Hyperparametrization		41.32%	40.60%	41.38%	40.43%
Preprocessing + TF-IDF + CV + Hyperparametrization		40.02%	43.43%	39.75%	38.81%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		39.97%	44.02%	39.71%	38.80%

Preprocessing +BoW	Random Forest	41.53%	40.71%	41.60%	40.39%
Preprocessing + TF-IDF		41.42%	40.85%	41.48%	40.44%
Preprocessing + TF-IDF (10000 keywords)		41.11%	40.16%	41.12%	40.16%
Preprocessing + BoW + CV + Hyperparametrization		39.58%	38.54%	39.68%	37.28%
Preprocessing + TF-IDF + CV + Hyperparametrization		39.30%	38.49%	39.39%	37.14%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		39.97%	39.11%	40.02%	37.94%
Preprocessing + BoW	XGBoost	44.64%	44.85%	44.63%	44.39%
Preprocessing + TF-IDF		44.28%	44.31%	44.25%	43.94%
Preprocessing + TF-IDF (10000 keywords)		43.23%	43.34%	43.23%	42.97%
Preprocessing +BoW	LightGBM	45.75%	45.82%	45.76%	45.41%
Preprocessing + TF-IDF		45.85%	45.77%	45.85%	45.61%
Preprocessing + TF-IDF (10000 keywords)		45.62%	45.52%	45.63%	45.37%
Preprocessing + BoW	CatBoost	45.62%	45.89%	45.67%	45.23%
Preprocessing + TF-IDF		45.75%	45.87%	45.77%	45.47%
Preprocessing + TF-IDF (10000 keywords)		46.19%	46.24%	46.22%	45.89%

5.1.3. Case 1.3: Preprocessing + 6 classes (major class balanced)

Lastly, an equivalent experiment was trialed with a balanced dataset and only the class with the most records decreased its number to equip the class with fewer records. The metric considered to evaluate the results of this Case was Accuracy. Half of the classifiers performed better in the experiments with BoW and the other half with TF-IDF. Still, the most significant increase was noted in the SVM and Logistic Regression models, where there was an increase of around 7 and 5 percentage points in the experiments with TF-IDF compared to BoW (Table 10). The other classifiers had similar results in the two experiments. The Decision Tree got the worst performance in all experiments once again and the LightGBM got an Accuracy of 47.77%, the highest.

In the experiments with TF-IDF and a limitation of ten thousand keywords, the Accuracy result was superior for all models except for the MNB, XGBoost, and LightGBM, which performed better in the BoW experiment. In the algorithms where Accuracy rose, the difference was not very marked, with MNB being the one that went up the most, nearly 6 percentage points, but failing to reach an Accuracy of 43.35% achieved in the experiment with BoW.

The application of hyperparameters to the algorithms was beneficial for most of the models, with LR achieving the best result in this Case, 47.91%, in the experiment with TF-IDF, limitation of ten thousand keywords, hyperparameters (*max_iter* = 10000, *C* = 1, *intercept_scaling* = 1, *penalty* = "l2", *solver* = "lbfgs", *tol* = 0.0001) and five-fold cross-validation. On the contrary, MNB and RF decreased their performance in all experiments with hyperparametrization and cross-validation, with the most notable decrease being 5 percentage points in the TF-IDF experiment with a limitation of the number of keywords.

Table 10 - Results of Case 1.3 (preprocessing + 6 classes with major class balanced).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + BoW	Decision Tree	29.94%	32.29%	29.39%	30.74%
Preprocessing + TF-IDF		31.25%	30.41%	30.12%	30.20%
Preprocessing + TF-IDF (10000 keywords)		31.73%	31.73%	30.94%	30.92%
Preprocessing + BoW + CV + Hyperparametrization		30.26%	32.60%	30.26%	31.82%
Preprocessing + TF-IDF + CV + Hyperparametrization		32.44%	31.62%	31.50%	31.50%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyper parametrization		32.19%	31.57%	31.35%	31.39%
Preprocessing + BoW	Support Vector Machine	37.83%	36.84%	36.68%	36.65%
Preprocessing + TF-IDF		44.36%	43.11%	42.95%	42.75%
Preprocessing + TF-IDF (10000 keywords)		44.41%	43.17%	43.00%	42.78%
Preprocessing + BoW + CV + Hyperparametrization		42.24%	41.38%	40.99%	40.95%
Preprocessing + TF-IDF + CV + Hyperparametrization		47.47%	46.90%	45.75%	45.23%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		47.61%	47.11%	45.82%	45.31%
Preprocessing + BoW	Logistic Regression	41.45%	40.66%	40.05%	40.16%
Preprocessing + TF-IDF		47.11%	46.49%	45.35%	45.36%
Preprocessing + TF-IDF (10000 keywords)		47.26%	46.63%	45.45%	45.45%
Preprocessing + BoW + CV + Hyperparametrization		46.19%	46.80%	43.74%	44.05%
Preprocessing + TF-IDF + CV + Hyperparametrization		47.70%	47.39%	46.05%	46.18%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		47.91%	47.51%	46.23%	46.32%
Preprocessing +BoW	Multinomial Naïve Bayes	43.35%	40.93%	41.72%	40.08%
Preprocessing + TF-IDF		34.84%	34.84%	28.58%	28.58%
Preprocessing + TF-IDF (10000 keywords)		40.67%	46.53%	36.12%	33.74%
Preprocessing + BoW + CV + Hyperparametrization		42.51%	41.51%	41.33%	39.28%
Preprocessing + TF-IDF + CV + Hyperparametrization		40.95%	42.37%	38.30%	36.28%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		35.13%	60.55%	34.21%	27.35%
Preprocessing +BoW	Random Forest	43.07%	42.68%	40.59%	39.05%

Preprocessing + TF-IDF		42.66%	42.07%	39.72%	38.31%
Preprocessing + TF-IDF (10000 keywords)		43.14%	42.05%	40.60%	39.22%
Preprocessing + BoW + CV + Hyperparametrization		38.55%	46.13%	34.43%	31.16%
Preprocessing + TF-IDF + CV + Hyperparametrization		37.59%	44.38%	33.05%	29.83%
Preprocessing + TF-IDF (10000 keywords) + CV + Hyperparametrization		40.30%	44.51%	36.44%	33.29%
Preprocessing + BoW	XGBoost	46.93%	46.17%	45.22%	44.92%
Preprocessing + TF-IDF		45.99%	45.99%	44.37%	44.37%
Preprocessing + TF-IDF (10000 keywords)		46.14%	45.54%	44.52%	44.40%
Preprocessing +BoW	LightGBM	47.77%	47.10%	46.17%	45.86%
Preprocessing + TF-IDF		46.97%	46.97%	46.97%	46.97%
Preprocessing + TF-IDF (10000 keywords)		46.85%	46.06%	45.34%	45.19%
Preprocessing + BoW	CatBoost	47.08%	47.41%	45.00%	45.01%
Preprocessing + TF-IDF		47.50%	47.50%	45.58%	45.58%
Preprocessing + TF-IDF (10000 keywords)		47.67%	47.49%	45.73%	45.71%

5.2. Case 2: No Preprocessing + 6 classes

In this scenario, the keywords' extraction technique chosen was TF-IDF since the results were more positive than those achieved in the Bag of Words experiments in Case 1. Two attempts were made, one with no feature limitation and another with only ten thousand keywords (Table 11). Once again, the Decision Tree classifier was denoted as the classifier with the worst performance, and Logistic Regression, unlike the experiments with preprocessing, got the best result with 50.76% Precision, which means it only got half of the labels correct. For these experiments, an imbalanced dataset was used. In the experiments with feature limitation, having the last experiences in mind, the results were slightly better in almost all classifiers as it was not expected, with Logistic Regression, the best classifier, having a precision of 51.16%. This classifier got the best performance of all.

Table 11- Results of Case 2 (without preprocessing).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Wo Preprocessing + TF-IDF	Decision Tree	30.06%	30.57%	27.66%	29.02%
Wo Preprocessing + TF-IDF (10000 keywords)		32.35%	32.35%	28.98%	28.98%
Wo Preprocessing + TF-IDF	Support Vector Machine	47.04%	46.15%	43.33%	44.14%
Wo Preprocessing + TF-IDF (10000 keywords)		47.01%	45.94%	45.94%	44.10%
Wo Preprocessing + TF-IDF	Logistic Regression	49.74%	50.76%	44.64%	46.24%
Wo Preprocessing + TF-IDF (10000 keywords)		50.03%	51.16%	51.16%	46.70%
Wo Preprocessing + TF-IDF	Multinomial Naïve Bayes	33.61%	37.17%	19.22%	12.51%
Wo Preprocessing + TF-IDF (10000 keywords)		38.54%	49.28%	26.05%	26.05%
Wo Preprocessing + TF-IDF	Random Forest	40.59%	48.25%	29.28%	28.69%
Wo Preprocessing + TF-IDF (10000 keywords)		42.49%	48.23%	32.21%	32.41%
Wo Preprocessing + TF-IDF	XGBoost	46.97%	47.17%	41.29%	42.56%
Wo Preprocessing + TF-IDF (10000 keywords)		47.77%	48.37%	42.17%	43.60%
Wo Preprocessing + TF-IDF	LightGBM	49.22%	49.80%	43.86%	45.23%
Wo Preprocessing + TF-IDF (10000 keywords)		49.45%	49.96%	44.07%	45.43%
Wo Preprocessing + TF-IDF	CatBoost	48.91%	49.96%	43.25%	44.76%
Wo Preprocessing + TF-IDF (10000 keywords)		48.81%	50.03%	43.32%	44.84%

5.3. Case 3: Preprocessing + 6 classes with prevalent words

To get better results - the number of features considerably changed – by considering only the ones that appear more often in each class. Thus, the model precision would be expected to improve as each complaint's topic got narrower, although the results from Table 12 show the contrary. The fact that the complaints had fewer words had a very negative impact on the performance of the models. Since, in the first Case, the experiments with TF-IDF got better results than the ones with BoW, TF-IDF was used to extract the keywords. For the same reason, the dataset with imbalanced classes was chosen.

The Support Vector Machine classifier had the best Precision with 40.27%, the worst result for this classifier considering the previous Cases with TF-IDF to select the keywords. The classifier with the poorest Precision result was the Decision Tree similar to the Cases analysed before, but in this experiment, it only achieved 26.59%, the lowest performance to this point. Although in Case 2, a second experiment was trialed, with the limitation of keywords to the thousand most relevant ones, in this Case, the results were considerably lower, and so other experiments were discarded.

Table 12 - Results of Case 3 with prevalent words.

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Keywords + Preprocessing + TF-IDF	Decision Tree	20.45%	26.59%	18.70%	21.88%
	Support Vector Machine	36.96%	40.27%	28.15%	25.34%
	Logistic Regression	37.58%	36.10%	29.24%	27.46%
	Multinomial Naïve Bayes	34.98%	28.81%	23.11%	19.43%
	Random Forest	35.33%	31.89%	29.15%	28.99%
	XGBoost	37.34%	34.85%	30.44%	30.44%
	LightGBM	37.72%	35.37%	30.97%	30.91%
	CatBoost	38.13%	35.54%	31.33%	31.19%

5.4. Case 4: Preprocessing + 3 classes

5.4.1. Case 4.1: Preprocessing + 3 imbalanced classes

Comparing the results of Case 1.1 with the results of the present Case, since in both cases preprocessing was applied, varying only the number of classes, it is noticeable that the Precision value rose in all experiments with different classifiers (Table 13). Despite the worst performance, the Decision Tree classifier had the highest increase of almost 16 percentual

points in the experiment without hyperparametrization and cross-validation. Overall, the Precision values vary between 55% and 57% Precision, with the Catboost classifier obtaining the best result: 57.85%. Unlike the previous Case with 6 classes, the experiments with hyperparameters did not optimize the results significantly, and in some cases, the performance was worse, as in the Random Forest example. However, there were no cases with high Precision and other evaluation metrics close to 20%. In Case 4, three new models were introduced, XGBoost, LightGBM and Catboost. Although the LightGBM model achieved the best result, it was not considerably superior to the results obtained with the other classifiers and only one experiment was performed.

Table 13 - Results of Case 4.1 (preprocessing + 3 imbalanced classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	46.62%	46.63%	46.25%	46.35%
Preprocessing + TF-IDF + CV + Hyperparametrization		47.32%	49.14%	48.88%	46.19%
Preprocessing + TF-IDF	Support Vector Machine	54.93%	55.00%	54.53%	54.67%
Preprocessing + TF-IDF + CV + Hyperparametrization		57.38%	57.75%	56.90%	57.12%
Preprocessing + TF-IDF	Logistic Regression	57.09%	57.37%	56.63%	56.84%
Preprocessing + TF-IDF + CV + Hyperparametrization		57.09%	57.37%	56.63%	56.84%
Preprocessing + TF-IDF	Multinomial Naïve Bayes	49.77%	55.72%	46.92%	43.57%
Preprocessing + TF-IDF + CV + Hyperparametrization		52.83%	55.71%	51.66%	51.36%
Preprocessing + TF-IDF	Random Forest	53.77%	55.44%	52.37%	52.32%
Preprocessing + TF-IDF + CV + Hyperparametrization		49.54%	53.35%	47.85%	45.67%
Preprocessing + TF-IDF	XGBoost	55.84%	55.90%	55.78%	55.64%
Preprocessing + TF-IDF	LightGBM	57.75%	57.85%	57.75%	57.63%
Preprocessing + TF-IDF	CatBoost	57.20%	57.34%	57.19%	57.01%

5.4.2. Case 4.2: Preprocessing + 3 balanced classes

For this Case, the same experiments as in Case 4.1 were run, and Accuracy was used to evaluate the results. Regarding the experiments with TF-IDF, most of the algorithms scored over 50%, as seen in Table 14, and the Decision Tree algorithm was the only one to underperform with 46.86% Accuracy. The model that achieved the best result in this experiment, and in general, was the Catboost: 58.38%, followed by the LightGBM and the Logistic Regression.

In the experiments with cross-validation and hyperparametrization, the results were better, except for the RF, as happened in the Case above, and the SVM was the algorithm that

improved the most, around 3 percentage points. Unlike Case 1, the experiment with balanced classes scored better than that with imbalanced classes.

Table 14 - Results of Case 4.2 (preprocessing + 3 balanced classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	46.86%	46.71%	46.88%	46.76%
Preprocessing + TF-IDF + CV + Hyperparametrization		47.19%	47.05%	47.19%	47.19%
Preprocessing + TF-IDF	Support Vector Machine	55.57%	55.50%	55.59%	55.45%
Preprocessing + TF-IDF + CV + Hyperparametrization		58.35%	58.31%	58.40%	58.14%
Preprocessing + TF-IDF	Logistic Regression	57.35%	57.30%	57.37%	57.22%
Preprocessing + TF-IDF + CV + Hyperparametrization		58.05%	57.96%	58.06%	57.91%
Preprocessing + TF-IDF	Multinomial Naïve Bayes	54.04%	54.39%	54.11%	53.81%
Preprocessing + TF-IDF + CV + Hyperparametrization		54.49%	55.16%	54.55%	53.84%
Preprocessing + TF-IDF	Random Forest	53.49%	53.41%	53.52%	53.34%
Preprocessing + TF-IDF + CV + Hyperparametrization		52.81%	53.55%	52.85%	52.16%
Preprocessing + TF-IDF	XGBoost	57.07%	57.09%	57.09%	56.71%
Preprocessing + TF-IDF	LightGBM	57.97%	57.93%	58.01%	57.74%
Preprocessing + TF-IDF	CatBoost	58.38%	58.61%	58.40%	58.09%

5.4.3. Case 4.3: Preprocessing + 3 classes (major class balanced)

A new experiment was performed with the major class balanced, and the behaviour of the models was similar to the two previous Cases. In the first experiment with TF-IDF, the Catboost algorithm achieved the best Accuracy with 58.56% and was the top result of all experiments (Table 15). The Decision Tree algorithm was once again the algorithm with the worst results, but, in this Case, it obtained the lowest results of Case 4.

By applying the hyperparameters chosen through grid-search and a five-fold cross-validation, the Accuracy has decreased in the RF model as in the last Case, but also in the MNB model. The SVM algorithm got the best Accuracy in this last experiment with 58.34% and went up almost 3 percentage points compared to the experiment without hyperparameters and CV.

Table 15 - Results of Case 4.3 (preprocessing + 3 classes with major class balanced).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	43.00%	47.07%	42.97%	44.93%
Preprocessing + TF-IDF + CV + Hyperparametrization		46.96%	46.85%	46.92%	46.87%
Preprocessing + TF-IDF	Support Vector Machine	55.53%	55.46%	55.50%	55.38%
Preprocessing + TF-IDF + CV + Hyperparametrization		58.34%	58.31%	58.30%	58.12%
Preprocessing + TF-IDF	Logistic Regression	57.81%	57.76%	57.76%	57.62%
Preprocessing + TF-IDF + CV + Hyperparametrization		57.81%	57.76%	57.76%	57.62%
Preprocessing + TF-IDF	Multinomial Naïve Bayes	54.60%	54.68%	54.57%	54.40%
Preprocessing + TF-IDF + CV + Hyperparametrization		53.68%	54.04%	53.58%	52.89%
Preprocessing + TF-IDF	Random Forest	53.88%	53.71%	53.84%	53.65%
Preprocessing + TF-IDF + CV + Hyperparametrization		52.67%	53.52%	52.57%	51.80%
Preprocessing + TF-IDF	XGBoost	57.05%	57.04%	57.00%	56.73%
Preprocessing + TF-IDF	LightGBM	58.13%	58.11%	58.08%	57.87%
Preprocessing + TF-IDF	CatBoost	58.56%	58.64%	58.50%	58.21%

5.4.4. Case 4.4: Preprocessing + 3 balanced classes + OVO and OVR

Regarding the experiment with OVO and OVR methods, it is possible to state that when the OVO experiment had better results than OVR, the values were very close, with the Decision Tree algorithm having the highest discrepancy of 2 percentage points, as observed in Table 16. The RF, the MNB, and the Catboost models scored the same in both experiments: 55%, 54% and 58%, respectively.

On the other hand, the classifiers that performed better in the OVR experiments, XGBoost and LightGBM, had a difference of about 10 percentage points from those obtained in the OVO experiments. Even though the LR and the LightGBM models also scored 58% Accuracy, the Catboost was considered the best algorithm because the results were consistent in both experiments.

Table 16 - Results of Case 4.4 (preprocessing + 3 balanced classes + OVO and OVR.

Experiments	Classifier	Accuracy
One-vs-One	Decision Tree	47.00%
One-vs-Rest		45.00%
One-vs-One	Support Vector Machine	56.00%
One-vs-Rest		55.00%
One-vs-One	Logistic Regression	58.00%
One-vs-Rest		57.00%
One-vs-One	Multinomial Naïve Bayes	54.00%
One-vs-Rest		54.00%
One-vs-One	Random Forest	55.00%
One-vs-Rest		55.00%
One-vs-One	XGBoost	47.00%
One-vs-Rest		57.00%
One-vs-One	LightGBM	47.00%
One-vs-Rest		58.00%
One-vs-One	Catboost	58.00%
One-vs-Rest		58.00%

5.5. Case 5: Transfer learning + 3 classes

Since the classifiers tested in the previous experiments did not reach satisfactory results, another experiment was made with a transfer learning BERT model. The performance of this model can be consulted in Table 17. As the results reached 54% with only 3 epochs due to the response capacity of the servers both locally and on Google Collab and Kaggle, it is possible to infer that the Accuracy would not go much higher with an increase in the number of epochs.

Table 17 - Results of Case 5 with a Transfer Learning model.

Experiments	Classifier	Accuracy
BERT	pedro-m4u / 1000_respostas-MODELO_2	54.00%

5.6. Case 6: Preprocessing + targeted classes

5.6.1. Case 6.1: Preprocessing + 3 targeted classes

To understand if the unsatisfactory results obtained from the previous experiments were because complaints were too long, the way of writing differed from person to person, or the classes were too broad, more specific ones have been adopted. By doing so, the results increased

substantially and were better than all the other Cases. The DT and the MNB models were the worst performers as they did not reach 75% Accuracy, but the RF algorithm was very close to 80%, and the others even exceeded it, with SVM reaching 85.08% Accuracy, the highest score (Table 18). However, it is important to recall that this dataset had a minimal number of complaints, only 603.

Table 18 - Results of Case 6.1 (preprocessing + 3 targeted classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	73.48%	74.47%	72.32%	72.93%
	Support Vector Machine	85.08%	85.47%	84.66%	84.97%
	Logistic Regression	84.53%	85.31%	84.06%	84.48%
	Multinomial Naïve Bayes	74.59%	75.85%	75.09%	74.86%
	Random Forest	77.90%	79.20%	77.59%	78.04%
	XGBoost	81.77%	83.42%	81.30%	81.98%
	LightGBM	80.11%	81.57%	79.62%	80.21%
	CatBoost	80.66%	82.29%	79.85%	80.53%

5.6.2. Case 6.2: Preprocessing + 5 targeted major classes

To rule out the hypothesis that the better results achieved in the last experiment were mainly due to the size of the dataset, a new one was created with the values that had the most complaints from each class. By adding two additional classes, the results decreased substantially. The Logistic Regression algorithm was the top scorer with an Accuracy of 70.58% (Table 19). However, this result was lower than the lowest result obtained in the last Case by the Decision Tree classifier.

The SVM, XGBoost and Catboost had very similar Accuracy results, around 70%, while the Multinomial Naïve Bayes model and the Decision Tree performed the worst, between 50% and 60%.

Table 19 - Results of Case 6.2 (preprocessing + 5 targeted major classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	50.75%	50.98%	50.97%	50.88%
	Support Vector Machine	68.55%	68.62%	68.67%	68.50%
	Logistic Regression	70.58%	70.81%	70.76%	70.68%
	Multinomial Naïve Bayes	59.85%	61.88%	60.19%	58.86%
	Random Forest	61.07%	60.60%	61.27%	60.64%
	XGBoost	66.98%	67.45%	67.15%	67.16%
	LightGBM	68.07%	68.62%	68.22%	68.23%
	CatBoost	68.23%	68.21%	68.21%	68.38%

5.6.3. Case 6.3: Preprocessing + 4 targeted classes

Since the previous results were lower than expected, the vaguer "Informações Prestadas" class was removed. The results presented in Table 20 show that eliminating a broader class - positively impacted the model's performance. The SVM and the LR algorithms got the same Accuracy, 80.73%, but the latter achieved better scores in the other evaluation metrics and was considered the best model.

The behaviour of the models was very similar compared with the last Case, as the Decision Tree and the MNB achieved the lowest results and the LR got the best performance. The results of XGBoost, LightGBM and Catboost were nearly identical.

Table 20 - Results of Case 6.3 (preprocessing + 4 targeted classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	64.86%	64.79%	65.07%	64.73%
	Support Vector Machine	80.73%	81.09%	80.92%	80.69%
	Logistic Regression	80.73%	81.32%	80.94%	80.68%
	Multinomial Naïve Bayes	70.20%	75.97%	70.80%	69.44%
	Random Forest	72.84%	73.43%	73.20%	72.54%
	XGBoost	77.25%	77.49%	77.47%	77.21%
	LightGBM	77.21%	78.64%	78.60%	78.27%
	CatBoost	77.25%	77.61%	77.50%	77.17%

5.6.4. Case 6.4: Preprocessing + 5 original classes

As the results in the last Case improved by eliminating one diverse class, another experiment was elaborated with the five original classes, excluding the *Client* class. Through the analysis of Table 21, it is possible to infer that the results were considerably low, since the Logistic Regression algorithm only achieved an Accuracy of 51.48% and was the best model of all, followed by the SVM. The Decision Tree classifier did not reach 40% and, together with MNB was the lowest-performing model. As for the other models, they scored closer to 50%.

Comparing these last results with those of Case 1 with six classes, they do not differ much, which may indicate that more diverse classes could negatively influence the results.

Table 21 - Results of Case 6.4 (preprocessing + 5 original classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	35.63%	34.95%	35.27%	35.08%
	Support Vector Machine	49.34%	49.05%	49.10%	49.07%
	Logistic Regression	51.48%	51.30%	51.29%	51.28%
	Multinomial Naïve Bayes	42.61%	47.34%	43.07%	41.17%
	Random Forest	48.93%	48.28%	48.64%	48.25%
	XGBoost	47.04%	46.86%	46.71%	46.75%
	LightGBM	48.77%	48.48%	48.45%	48.39%
	CatBoost	48.93%	48.37%	48.48%	48.33%

5.6.5. Case 6.5: Preprocessing + 4 original classes

By eliminating the *Other* class, the Accuracy increased in all models. Despite having the lowest performance, the Decision Tree and the MNB algorithms registered the largest increase of more than 8 percentual points (see Table 22). The best score was achieved by the Logistic Regression classifier, once again, with 57.44% of Acuracy. The LightGBM model had the smallest oscillation since it only increased around 4 percentual points.

These results reinforce the theory that the more specific the classes are, the more accurate the classification is.

Table 22 - Results of Case 6.5 (preprocessing + 4 original classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	44.00%	43.57%	43.44%	43.47%
	Support Vector Machine	57.23%	56.38%	56.65%	56.47%
	Logistic Regression	57.44%	56.63%	56.98%	56.72%
	Multinomial Naïve Bayes	50.97%	54.67%	50.87%	51.50%
	Random Forest	51.38%	51.31%	50.90%	50.88%
	XGBoost	54.77%	55.80%	54.54%	54.78%
	LightGBM	52.92%	53.76%	52.62%	52.78%
	CatBoost	56.00%	56.08%	55.65%	55.53%

5.6.6. Case 6.6: Preprocessing + 3 original classes

In the last Case, as expected, the results rose with the elimination of another broader class. Unlike the previous Cases, the Support Vector Machine got the best performance with 71.41% Accuracy followed by the Catboost and the Logistic Regression models, which also scored over 70% (see Table 23) . Considering all the Cases experimented, the most effective method was to eliminate classes whose topics could be applied to other classes.

Table 23 - Results of Case 6.6 (preprocessing + 3 original classes).

Experiments	Classifier	Accuracy	Precision	Recall	F1-Measure
Preprocessing + TF-IDF	Decision Tree	54.86%	54.58%	54.52%	54.53%
	Support Vector Machine	71.41%	71.09%	71.12%	71.09%
	Logistic Regression	71.00%	70.73%	70.72%	70.70%
	Multinomial Naïve Bayes	64.98%	68.43%	65.50%	64.90%
	Random Forest	66.76%	66.83%	66.75%	66.52%
	XGBoost	69.63%	69.69%	69.51%	69.43%
	LightGBM	69.22%	69.08%	69.12%	68.97%
	CatBoost	71.14%	71.09%	71.02%	70.94%

Conclusion

This study aimed to present a solution to reduce the manual tasks performed by Worten employees in classifying complaints through a data-driven approach. In order to do this, it was necessary to gather relevant literature on this topic, through scientific Articles and Conference Papers. The results showed that no solutions were developed to address these challenges in electronics retail companies, demonstrating a gap and accentuating the relevance of this study. After broadening the research topic, it became clear that there is no consensus in the preprocessing and keyword extraction techniques applied and in the ML models since there is a great deal of variety.

The dataset used in this study not only had many irrelevant columns but also complaints that were meaningless to classify, and eliminating them contributed to a drastic reduction in the number of records, around 40.3%. Once preprocessing was applied and cases with missing values were dealt with, it was essential to understand the distribution of the values for each attribute as well as the content of the complaints through different analysis. The clusters created in the different scenarios were hardly distinguishable from the others, demonstrating the disparity of terms used by each customer when complaining, which made the classification task particularly challenging.

Different experiments emerged in an attempt to improve the previous one, reaching a total of sixteen, with different datasets, classes and methods. Considering the three experiments of Case 1, the main aspect they have in common is that the Decision Tree classifier had the worst performance in all of them. In the experiments with equally balanced classes and only the majority class balanced, TF-IDF achieved a better result than BoW in most classifiers, which did not happen in the experiment with the imbalanced classes. The implementation of word limitation with TF-IDF was only effective in the experiment with the major class balanced and the best results in each of the Cases were obtained by different classifiers and experiments. In the first Case, the SVM got the top Precision, 50.46%, in the experiment with TF-IDF, hyperparametrization, and a five-fold CV, while in the remaining Cases, LR was the best classifier in the experiments with TF-IDF and TF-IDF considering the ten thousand most relevant features, the implementation of hyperparameters and cross-validation. It is noticeable that the equal balance of the classes worsened the results by almost 4 percentage points and that the Case with the major class balanced slightly improved performance with an Accuracy of 47.91% but did not reach the result of Case 1.1 with the imbalanced classes.

In the Case without preprocessing applied, the results improved slightly since the Logistic Regression model in the experiment with TF-IDF and a limitation of ten thousand features achieved an Accuracy of 51.16%. Since the difference in results compared to the best score in Case 1 was less than 1 percentage point, the following experiments were all carried out with preprocessing, following other studies in the same domain discussed in the Literature Review Chapter.

Of all the experiments, the one with the worst results was the Case with the most predominant words in each class. This Case emerged because the pie charts in the Data Understanding Chapter showed that the words that appeared most often in each class were the same as those in other classes. However, when analysing the results, it was clear that the model's behavior did not meet expectations, as the best result of Precision achieved by the SVM was only 40.27%. This may be because the words that distinguish between classes are not necessarily the most common in each class.

The reduction from six classes to three was carried out in order to improve the classification task by combining similar classes; however, the improvement was not that significant since it fell short of 10 percentage points. Unlike Case 1, the experiment with imbalanced classes performed better than the experiments with equally balanced classes in almost all classifiers, but worse than the Case with the major class balanced. Regarding the best results in each Case, the experiment with imbalanced classes had the lowest performance, 57.85% of Precision, with the LightGBM classifier. The remaining had very similar results, with Catboost being the best classifier in both experiments, in which the experiment with the major class balanced stood out with 58.56% Accuracy. The experiments with the OVR and OVO methods did not turn out to be very beneficial since the results were inferior to the three previous Cases.

The only experiment with a Transfer Learning model, BERT, proved to be computationally heavy and the results did not surpass the previous ones, since they did not exceed 50%. The results could possibly be better with more powerful machines, but the 54% Accuracy achieved with three epochs does not suggest a much better result.

Lastly, Case 6 confirmed that the more specific and distinct the classes, the better the results. The performance of the models, even with a much smaller sample than in the previous Cases, rose by almost 20 percentage points, and the best result was achieved by the SVM algorithm with 85.08% of Accuracy. The following two experiments, with more complaints, showed that as a class with vaguer values was removed, the performance of the models was positively impacted. Since these first three experiments had fewer records, they served as a test for the following experiments with more substantial classes. Thus, this method was applied to

the original classes. By eliminating the Client class, the Logistic Regression model achieved an Accuracy of 51.48%, representing an increase of approximately 1 percentage point compared to Case 1. The elimination of the *Other* class improved the previous results, with an accuracy of 57.44%, an approximate result to that obtained in Case 4 with only 3 classes. Finally, the classification of *Money*, *Technical Problem* and *Time* classes, by eliminating the *Service* class, resulted in the best result achieved, 71.41% of Accuracy with the Support Vector Machine model.

In conclusion, the TF-IDF achieved better results than BoW and the limitation of keywords did not have a great impact in the performance of the models, as well as the balancing of the classes. Despite having slightly better results, the experiment without preprocessing did not prove to be a decisive factor in improving performance, and the reduction of the complaints text to specific keywords only worsened the results, as it was the worst Case of all. Reducing the number of classes by half contributed to better results, despite not reaching 60%, and the results of the Transfer Learning approach, although indicative, cannot be considered realistic due to the lack of computing power. In the end, the most effective method was eliminating successive classes until they were as distinct as possible. The Decision Tree algorithm was the lowest performer in all Cases, while the LR achieved the best result in almost half the Cases, followed by SVM and Catboost. Nevertheless, the SVM performed the best Accuracy, 71.41%, in the Case of the three targeted classes.

This study has fulfilled the purpose of classifying complaints automatically; however, the proposed solution will not completely eliminate the manual work done by employees yet, since there are still complaints that have not been classified correctly and human verification is required. For future work, it would be recommended to test different stemmers and lemmatizers for the Portuguese language to measure their impact on preprocessing and subsequently on model performance. As the different results demonstrate that the classes' content is essential in this task, it would be meaningful to test different combinations of the 118 values in the *Reason* attribute, group the classes differently, and test them in more robust models that require greater computing power. The fact that few scientific articles deal with this topic suggests a great opportunity for more approaches to be tested.

References

Miquel-Romero, M. J., Frasquet, M., & Molla-Descals, A. (2020). The role of the store in managing postpurchase complaints for omnichannel shoppers. Journal of Business Research, 109, 288-296.

Vollero, A., Sardanelli, D., & Siano, A. (2021). Exploring the role of the Amazon effect on customer expectations: An analysis of user-generated content in consumer electronics retailing. Journal of Consumer Behaviour.

Jeanpert, S., Jacquemier-Paquin, L., & Claye-Puaux, S. (2021). The role of human interaction in complaint handling. Journal of Retailing and Consumer Services, 62, 102670.

Frasquet, M., Ieva, M., & Ziliani, C. (2019). Understanding complaint channel usage in multichannel retailing. Journal of Retailing and Consumer Services, 47, 94-103.

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., & Flach, P. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. IEEE Transactions on Knowledge and Data Engineering, 33(8), 3048-3061.

Bowen, J. P., & Giannini, T. (2014). Digitalism: the new realism? Electronic Visualisation and the Arts (EVA 2014), 324-331.

Watson, B. C. (2011). Barcode empires: politics, digital technology, and comparative retail firm strategies. Journal of industry, competition and trade, 11(3), 309-324.

Arkenback-Sundström, C. Technology and Information Technology in Work-based Training of Salespersons–A Retrospective Study of Retail Checkout Training and Learning.

Hagberg, J., Sundstrom, M., & Egels-Zandén, N. (2016). The digitalization of retailing: an exploratory framework. International Journal of Retail & Distribution Management.

Gupta, M. B., & Hooda, M. A. (2016). Retailing to E-Tailing: Evolution to Revolution. International journal of retail and marketing, 61-75.

Ramazanov, I. A., Panasenko, S. V., Cheglov, V. P., Krasil'nikova, E. A. E., & Nikishin, A. F. (2021). Retail Transformation under the Influence of Digitalisation and Technology Development in the Context of Globalisation. Journal of Open Innovation: Technology, Market, and Complexity, 7(1), 49.

Sadq, Z. M., Sabir, H. N., & Saeed, V. S. H. (2018). Analyzing the Amazon success strategies. Journal of process management. New Technologies, 6(4).

Özyirmidokuz, E. K., & Özyirmidokuz, M. H. (2014). Analyzing customer complaints: A web text mining application. In Proceedings of INTCESS14-International Conference on Education and Social Sciences, Istanbul (pp. 734-743).

Behere, T., Vaidya, A., Birhade, A., Shinde, K., Deshpande, P., & Jahirabadkar, S. (2020, July). Text summarization and classification of conversation data between service chatbot and customer. In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) (pp. 833-838).

Naresh, E., Vijaya Kumar, B., Pruthvi, V., Anusha K., Akshatha, V., (2019, December). Survey on Classification and Summarization of Documents. In 2019 Proceedings of the Second International Conference on Emerging Trends in Science & Technologies For Engineering Systems (ICETSE-2019).

Omurca, S. İ., Ekinci, E., Yakupoğlu, E., Arslan, E., & Çapar, B. Automatic Detection of the Topics in Customer Complaints with Artificial Intelligence. Balkan Journal of Electrical and Computer Engineering, 9(3), 268-277.

Lee, C. H., & Cranage, D. A. (2014). Toward understanding consumer processing of negative online word-of-mouth communication: the roles of opinion consensus and organizational response strategies. Journal of Hospitality & Tourism Research, 38(3), 330-360.

Harzing, A. W., & Alakangas, S. (2016). Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison. Scientometrics, 106(2), 787-804.

Martín-Martín, A., Orduna-Malea, E., Thelwall, M., & López-Cózar, E. D. (2018). Google Scholar, Web of Science, and Scopus: A systematic comparison of citations in 252 subject categories. Journal of informetrics, 12(4), 1160-1177.

Singh, R. P., Haque, R., Hasanuzzaman, M., & Way, A. (2020, December). Identifying complaints from product reviews in low-resource scenarios via neural machine translation. Association for Computational Linguistics (ACL).

Lee, S. H., Levin, D., Finley, P. D., & Heilig, C. M. (2019). Chief complaint classification with recurrent neural networks. Journal of biomedical informatics, 93, 103158.

Fahrudin, T., Buliali, J. L., & Fatichah, C. (2019). Ina-BWR: Indonesian bigram word rule for multi-label student complaints. Egyptian Informatics Journal, 20(3), 151-161.

Ali, M., Guru, D. S., & Suhil, M. (2018, December). Classifying Arabic Farmers' Complaints Based on Crops and Diseases Using Machine Learning Approaches. In International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 416-428).

Singh, G., Kumar, B., Gaur, L., & Tyagi, A. (2019, April). Comparison between multinomial and Bernoulli naïve Bayes for text classification. In 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (pp. 593-596).

Ali, M., Guru, D. S., & Suhil, M. (2018, December). Classifying Arabic Farmers' Complaints Based on Crops and Diseases Using Machine Learning Approaches. In International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 416-428).

Peng, X., Li, Y., Si, Y., Xu, L., Liu, X., Li, D., & Liu, Y. (2022). A social sensing approach for everyday urban problem-handling with the 12345-complaint hotline data. Computers, Environment and Urban Systems, 94, 101790.

Goncarovs, P. (2019, October). Active learning svm classification algorithm for complaints management process automatization. In 2019 60th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS) (pp. 1-3).

Liu, Y., Jiang, C., & Zhao, H. (2019). Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media. Decision Support Systems, 123, 113079.

HaCohen-Kerner, Y., Dilmon, R., Hone, M., & Ben-Basan, M. A. (2019). Automatic classification of complaint letters according to service provider categories. Information Processing & Management, 56(6), 102102.

Matos, B. C., Santos, R. B., Carvalho, P., Ribeiro, R., & Batista, F. (2022). Comparing different approaches for detecting hate speech in online Portuguese comments. In 11th Symposium on Languages, Applications and Technologies (SLATE 2022). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Freitas, C., & de Souza, E. (2023). A study on methods for revising dependency treebanks: in search of gold. Language Resources and Evaluation, 1-21.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages.

McArthur, J. J., Shahbazi, N., Fok, R., Raghubar, C., Bortoluzzi, B., & An, A. (2018). Machine learning and BIM visualization for maintenance issue classification and enhanced data collection. Advanced Engineering Informatics, 38, 101-112.

Fatima, S., & Srinivasu, B. (2017). Text Document categorization using support vector machine. International Research Journal of Engineering and Technology (IRJET), 4(2), 141-147.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, 5, 1-16.

Bangyal, W. H., Qasim, R., Rehman, N. U., Ahmad, Z., Dar, H., Rukhsar, L., & Ahmad, J. (2021). Detection of fake news text classification on COVID-19 using deep learning approaches. Computational and mathematical methods in medicine, 2021, 1-14.

Pratama, T., & Purwarianti, A. (2017, August). Topic classification and clustering on Indonesian complaint tweets for bandung government using supervised and unsupervised learning. In 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA) (pp. 1-6).

Qi, Z. (2020, June). The text classification of theft crime based on TF-IDF and XGBoost model. In 2020 IEEE International conference on artificial intelligence and computer applications (ICAICA) (pp. 1241-1246).

Tang, C., Luktarhan, N., & Zhao, Y. (2020). An efficient intrusion detection method based on LightGBM and autoencoder. Symmetry, 12(9), 1458.

Gu, Q., Sun, W., Li, X., Jiang, S., & Tian, J. (2023). A new ensemble classification approach based on Rotation Forest and LightGBM. Neural Computing and Applications, 35(15), 11287-11308.

Nandy, S., & Kumar, V. (2021, December). My Mind is a Prison: A Boosted Deep Learning approach to detect the rise in depression since COVID-19 using a stacked bi-LSTM CatBoost model. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 4396-4400).

Aldania, A. N. A., Soleh, A. M., & Notodiputro, K. A. (2023). A Comparative Study of CatBoost and Double Random Forest for Multi-class Classification. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 7(1), 129-137.

Khadhraoui, M., Bellaaj, H., Ammar, M. B., Hamam, H., & Jmaiel, M. (2022). Survey of BERT-base models for scientific text classification: COVID-19 case study. Applied Sciences, 12(6), 2891.

González-Carvajal, S., & Garrido-Merchán, E. C. (2020). Comparing BERT against traditional machine learning text classification.

Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. Journal of healthcare engineering, 2022.

Jia, B. B., Liu, J. Y., Hang, J. Y., & Zhang, M. L. (2023). Learning label-specific features for decomposition-based multi-class classification. Frontiers of Computer Science, 17(6), 1-10.

Appendix

Appendix A – Clusters generated from the values of the different dataset attributes

Figure A1 - Clusters generated from the *Reason* attribute.

Figure A2 - Clusters generated by the attribute *Category*.

Figure A3 - Clusters generated from the *Brand* variable.

Figure A4 - Clusters originated from the *Business Unit* variable.

Figure A5 - Clusters originated from the *Date* attribute.

Appendix B – Bar charts with the distribution of the top five values for each attribute

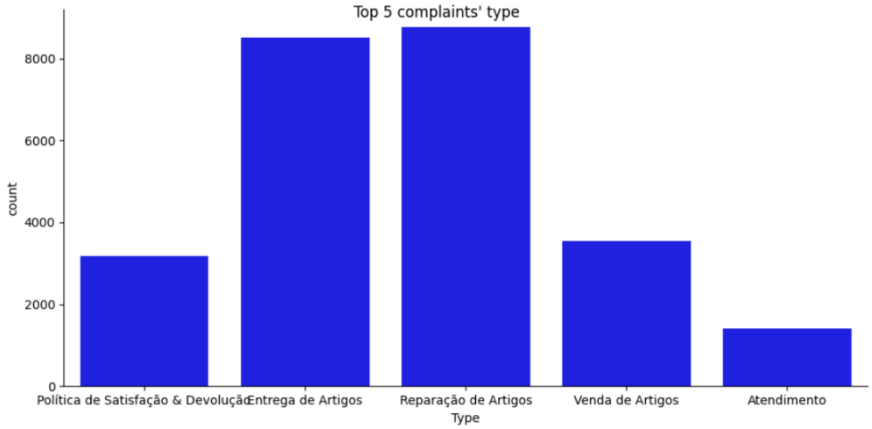

Figure B1 - Distribution of the five most common *Types* of complaint.

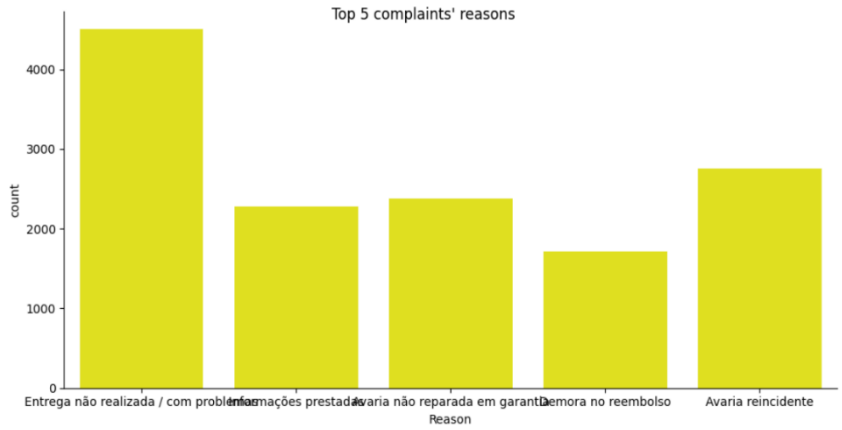

Figure B2 - Distribution of the 5 most common *Reasons* of complaint.

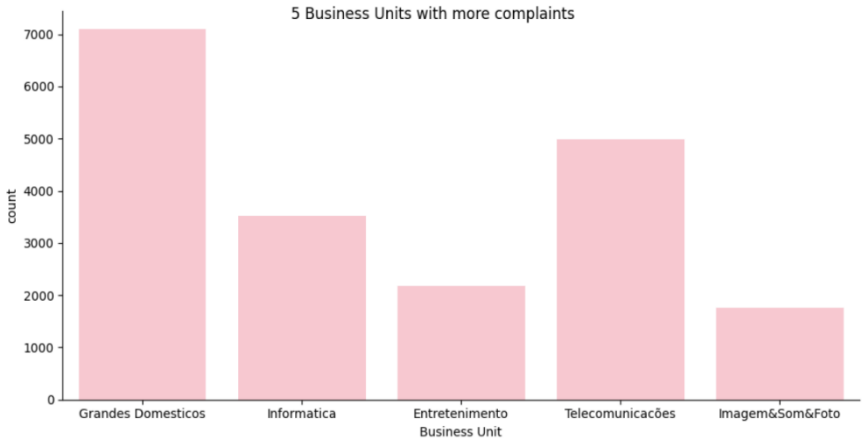

Figure B3 - Distribution of the top five *Business Units* with more complaints.

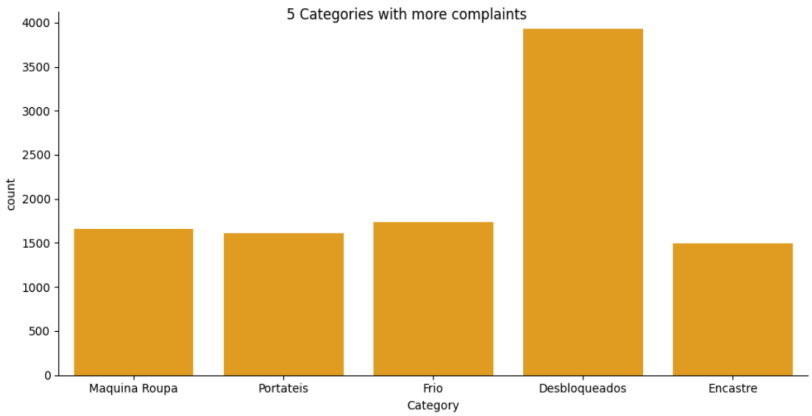

Figure B4 - Distribution of top five *Categories* with more complaints.

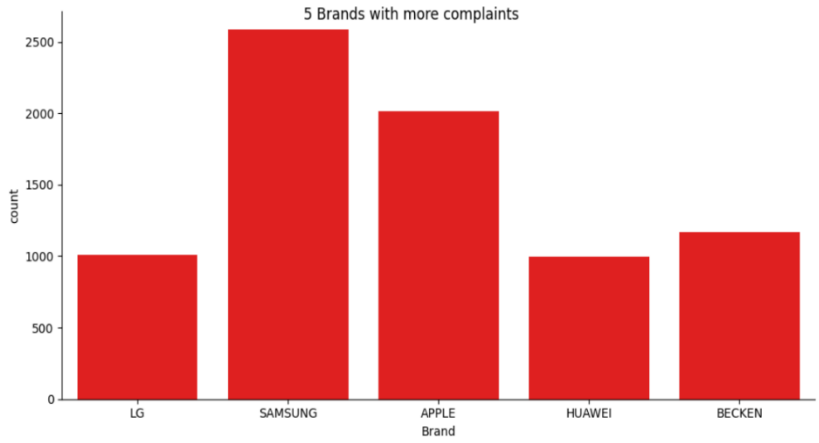

Figure B5 - Distribution of top five *Brands* with more complaints.

Appendix C – Split of the values of the *Reason* attribute into the different classes

Table C1 - Reason values that constitute the *Service* class.

Class	Reason
Service	Entrega não realizada / com problemas
	Avaria não reparada em garantia
	Entrega de artigo s/ condições
	Instalação Mal Efetuada / Não Efetuada
	Qualidade do serviço
	Serviço não realizado
	Serviço incorretamente realizado
	Danos provocados durante a instalação
	Entrega de acessório incorreto / trocado
	Serviço indisponível em loja
	Danos provocados durante o serviço
	Serviço incorretamento realizado
	Não disponibilização artigo empréstimo

Table C2 - Reason values that constitute the *Client* class.

Class	Reason
Client	Informações prestadas
	Informação incorreta
	Falta de informação
	Demora no atendimento
	Atendimento prioritário
	Inexistência de área de espera para Clientes

Table C3 - Reason values that constitute the *Technical Problem* class.

Class	Reason
Technical Problem	Avaria reincidente
	Rutura de Stock
	Danos provocados durante a reparação
	Danos durante a entrega
	Indisponibilidade dos sistemas de suporte de venda
	Artigo com sinais de utilização
	Falta de qualidade
	Não existe acessório
	Produto com aparência afetada
	Danos colaterais
	Produto com funcionamento indesejado/limitado
	Produto incompleto
	Fraca qualidade dos componentes
	Equipamentos desaquados
	Artigo provocou danos em casa, após uso
	Artigo potencialmente perigoso
	Produtos com embalagens abertas/ com dano
	Mobiliário causou danos nos Clientes

Table C4 - Reason values that constitute the *Money* class.

Class	Reason
Money	Demora no reembolso
	Política de Preço Mínimo Garantido
	Demora no envio da fatura
	Custos adicionais
	Diferença Preço entre Lojas
	Mensalidades
	Funcionalidade Pagamento
	Comissões/ Juros
	Oscilação de preços durante a campanha
	Passagem Garantia a Orçamento após serviço
	Orçamento maior que o valor do artigo
	Não aceita devolução através do mesmo meio de pagamento
	Não aceitação do modo de pagamento
	Pagamento por multibanco em dobro
	Regras Acumulação de Descontos
	Saldo Cartão Dá
	Código Promocional Online Inválido
	Pagamento por multibanco indisponível
	Preço elevado
	Impossibilidade de uso de código promocional em loja
	Não entrega de fatura de pagamento extra
	Preço

Table C5 - Reason values that constitute the *Time* class.

Class	Reason
Time	Falha na data da entrega
	Demora na resolução do processo
	Incumprimento de prazo de reparação
	Agendamento não realizado
	Incumprimento de prazo
	Demora no agendamento de entrega
	Entrega efetuda fora do período acordado
	Prazo de devolução expirado
	Demora na prestação do serviço
	Agendamento não cumprido
	Demora no envio da oferta
	Oferta não entregue
	Demora na entrega
	Demora na recolha do artigo
	Atraso no envio da oferta
	Não aceita prazo de reparação indicado
	Atraso na Instalação
	Agendamento não realizado
	Instalação não efetuada
	Prazo de validade do Cartão Dá
	Validade do Código Promocional Online
	Demora na comunicação dos vencedores

Table C6 - Reason values that constitute the *Other* class.

Class	Reason
Other	Artigo fora política satisfação/ devolução
	Não aceita política de satisfação/ devolução
	Cliente recusa a reparação do artigo
	Atitude comportamental
	Ausência de resolução
	Publicidade / Informação enganosa
	Impossibilidade de adesão
	Recolha de artigo usado
	Seguro declinado
	Problemas com a adesão
	Artigo não corresponde às expectativas do Cliente
	Não aceita artigo enviado da seguradora
	Compras não associadas no Cartão Resolve
	Regulamento pouco claro
	Devoluções por desistência
	Artigo de empréstimo durante a reparação
	Devolução não aceite por falta de talão
	Características do artigo
	Cliente não descreve motivo da insatisfação
	Certificação energética
	Devolução de serviço não aceite
	Cancelamento do serviço
	Impossibilidade de uso do Cartão Dá no site
	Manual de instruções
	Ausência de manual de instruções
	Ausência de promoção de boas práticas ambientais
	Artigo excluído
	Furtos
	Perda de Cartão Dá
	Arrumação/ Limpeza
	Não concorda com os vencedores
	Não aquisição dos bilhetes
	Não aceitação de resíduos obrigatórios por lei
	Temperatura
	Qualidade do ar
	Pavimento danificado

Appendix D – Pie charts displaying the most common words in each class

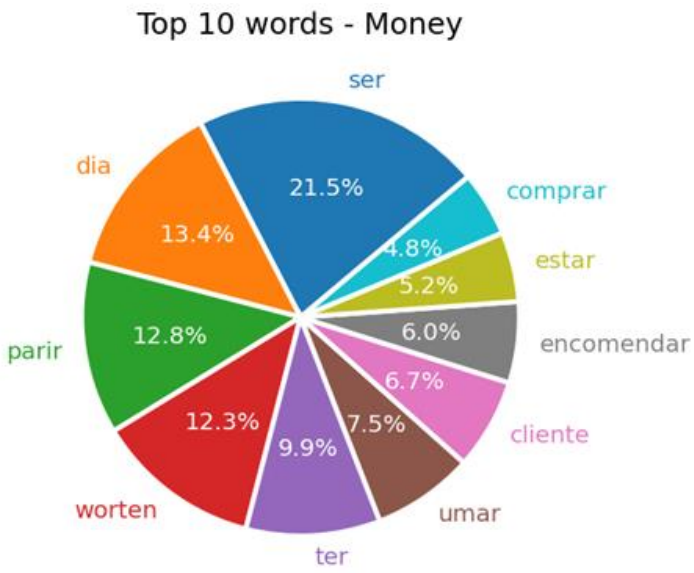

Figure D1 - Distribution of the ten most recurrent words in the *Money* class.

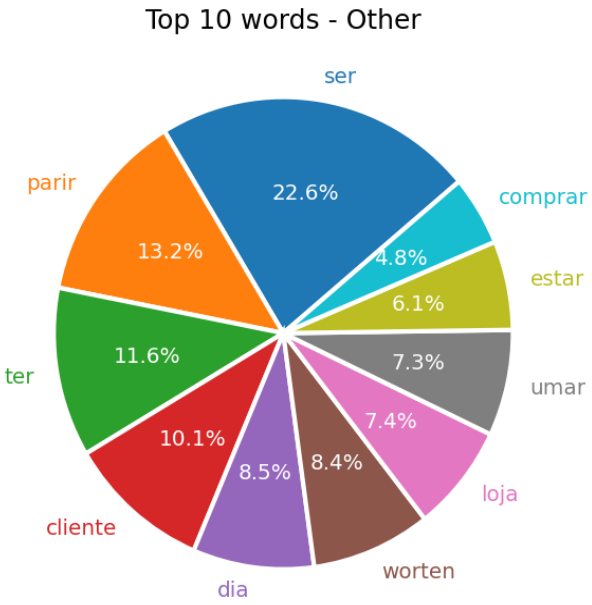

Figure D2 - Distribution of the ten most common words in the *Other* class.

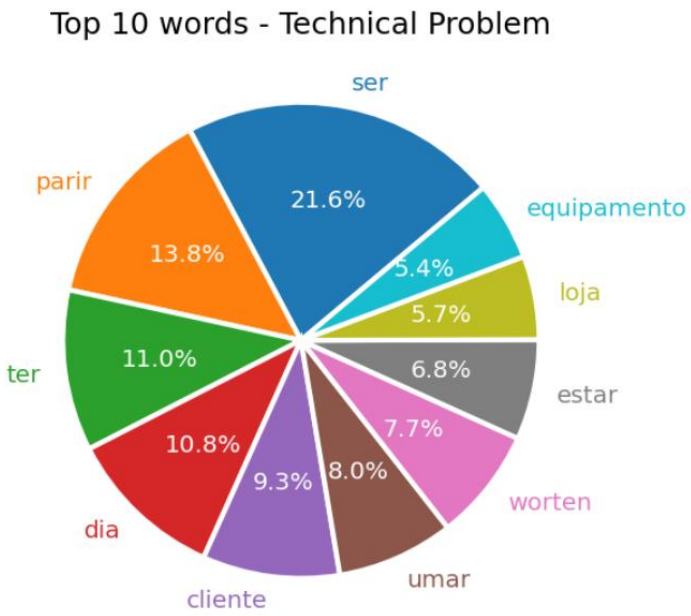

Figure D3 - Distribution of the top ten words in the *Technical Problem* class.

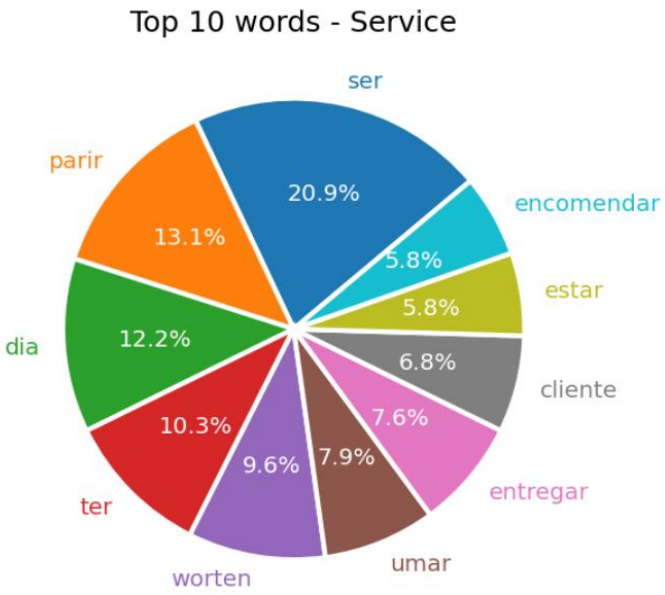

Figure D4 - Distribution of the ten most common words in the *Service* class.

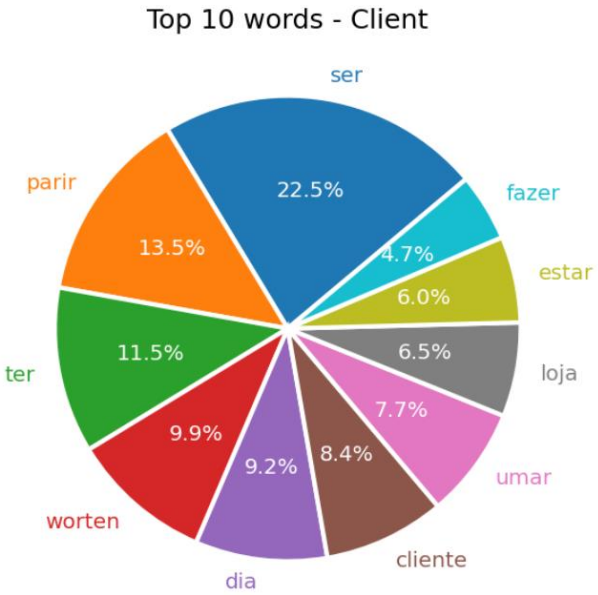

Figure D5 - Distribution of the top ten words in the *Client* class.