Product Complaint Understanding using NLP Techniques

Beatriz Marques Arcipreste
Mestrado em Ciência de Dados
Departamento de Ciência de Computadores
2023

Orientador
Evelin Carvalho Freire de Amorim, Professor Auxiliar Convidado,
Faculdade de Ciências da Universidade do Porto
Coorientador
Alípio Mário Guedes Jorge, Professor Catedrático,
Faculdade de Ciências da Universidade do Porto

U. PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Acknowledgements

I would like to express my gratitude to all of those who have contributed to the conclusion of this thesis.

Firstly, I'd like to thank my family for their love, encouragement, and belief in my abilities. Their support, both emotional and financial, has been a fundamental source of support throughout my journey.

Secondly, I would like to thank Professor Evelin Amorim and Professor Alípio Jorge for their guidance, invaluable insights, and knowledge throughout this research.

Finally, I want to thank my friends for encouraging me and reminding me that life is not only defined by professional achievements.

Thank you,

Beatriz Arcipreste

Abstract

Customer feedback is available through multiple channels: customer support services, phone calls, emails, apps, product websites, online forums, and social media. Most of this feedback is in the form of unstructured text which requires analytical tools such as Natural Language Processing (NLP) to extract useful insights that can be used by businesses to improve their product or service. The *Portal da Queixa* is an online platform where people can express their opinions and complaints on products, services, and brands. Given the large number of complaints generated daily, a system capable of predicting the reason of the complaint could decrease customer service spending and time and increase customer satisfaction, since the customer is redirected to the right department.

In this study, we investigate the issue of automatic complaint classification in the Portuguese language. The dataset used for training and evaluation consists of a collection of customer complaints from *Portal da Queixa* platform. The dataset analyzed in this work is highly imbalanced. To achieve our goals, we followed the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. We explore the use of Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings, such as Word2Vec and GloVe. During modelling, we explore both traditional machine learning algorithms and deep learning techniques. We explore multiple techniques to improve the performance of text classification systems such as SMOTE, class weighting, hyperparameter tuning, and feature selection. We use model's coefficients, SHAP, and error analysis to extract insights about the main issues faced by customers, and the quality of the labeling that is available.

The results of our experiments demonstrate that LinearSVC achieves competitive performance when compared to BERTimbau, achieving a macro F1-score of 0.50, while BERTimbau achieved 0.51 on the test set.

This thesis contributes to the field of Portuguese text classification by providing a comprehensive study of different techniques and approaches to improve text classification systems performance. The insights gained from this study can also help *Portal da Queixa* improve their labeling scheme.

Keywords: Customer Complaints, Text Classification, Imbalanced Data, CRISP-DM, Machine Learning, TF-IDF, Word Embeddings, BERTimbau

Resumo

A opinião do consumidor está disponível através de vários canais: serviços de apoio ao cliente, chamadas telefónicas, emails, aplicações, websites, fóruns online e redes sociais. A maioria deste feedback é apresentada sob a forma de texto não estruturado, o que requer ferramentas analíticas como o Processamento de Linguagem Natural (PLN) para extrair informações úteis que possam ser usadas pelas empresas para melhorar os seus produtos ou serviços. O Portal da Queixa é uma plataforma online onde as pessoas podem expressar as suas opiniões e reclamações sobre produtos, serviços e marcas. Dado o grande número de reclamações geradas diariamente, um sistema capaz de prever a razão da reclamação poderia diminuir os gastos com serviço de apoio ao cliente e aumentar a satisfação do cliente.

Este estudo, foca-se no problema da classificação automática de reclamações na língua portuguesa. O conjunto de dados usado consiste num conjunto de reclamações de clientes da plataforma do Portal da Queixa. Este conjunto de dados é bastante desequilibrado. Para atingir os nossos objetivos, seguimos o framework CRISP-DM (Cross-Industry Standard Process for Data Mining). Exploramos o uso da Frequência do Termo – Frequência Inversa dos Documentos (TF-IDF) e de representações de palavras, como o Word2Vec e o GloVe. Durante a modelação, exploramos tanto algoritmos tradicionais de aprendizagem automática como técnicas de aprendizagem profunda. Investigamos várias técnicas para melhorar o desempenho dos sistemas de classificação de texto, como SMOTE, ponderação de classes, otimização de hiperparâmetros e seleção de variáveis. Utilizamos os coeficientes do modelo, o SHAP e a análise de erros para extrair informações sobre os principais motivos de queixa, e sobre a qualidade da rotulagem disponível.

Os resultados demonstram que o LinearSVC alcança um desempenho competitivo quando comparado com o BERTimbau, alcançando um macro F1-Score de 0,50, enquanto o BERTimbau alcançou 0,51 no conjunto de teste.

Esta tese contribui para o campo da classificação de texto em língua portuguesa, fornecendo um estudo abrangente de diferentes técnicas para melhorar o desempenho dos sistemas de classificação de texto. As conclusões obtidas a partir deste estudo também podem ajudar o Portal da Queixa a melhorar o seu esquema de rotulagem.

Palavras-chave: Opinião do Consumidor, Classificação de Texto, Dados Desequilibrados, CRISP-DM, Aprendizagem Automática, TF-IDF, Representações de Palavras, BERTimbau

Table of Contents

List of Tables

List of Figures

Acronyms

ADASYN Adaptive Synthetic Sampling

BERT Bidirectional Encoder Representations from Transformers

BOW Bag of Words

CNN Convolutional Neural Network

CRISP-DM Cross-Industry Standard Process for Data Mining

DL Deep Learning

KNN k-Nearest Neighbors

LIME Local Interpretable Model-agnostic Explanations

LR Logistic Regression

LSTM Long Short-Term Memory

ML Machine Learning

NER Named Entity Recognition

NLP Natural Language Processing

ROS Random Oversampling

RUS Random Undersampling

SGD Stochastic Gradient Descent

SHAP SHapley Additive exPlanations

SMOTE Synthetic Minority Over-sampling Technique

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document Frequency

LINEARSVC Linear Support Vector Classifier

XGBOOST Extreme Gradient Boosting

Chapter 1

Introduction

In this introductory chapter, we present our motivation, the research problem, objectives, methodology, and structure of the thesis, as well as the expected contributions to the field of complaint analysis.

1.1 Context

Customer feedback is available through multiple channels: customer support services, phone calls, emails, apps, product websites, online forums, and social media. Most of this feedback is in the form of unstructured text which requires analytical tools such as Natural Language Processing (NLP) to extract useful insights that can be used by businesses to improve their product or service. Unfortunately, it is very challenging to work with customer feedback data for several reasons: customers use diverse language and expressions to express their opinions, the texts do not follow a particular structure, different industries have their terminology and the texts can contain irrelevant information.

Natural Language Processing results from the combination of different fields such as Linguistics, Computer Science, and Artificial Intelligence and it is concerned with creating systems that can understand written and spoken human language the same way as humans do. In this context, understanding unstructured text means extracting not only the semantics of the text but also the context and intention of the message.

Over the past years, there has been a big progress in NLP tasks, with the introduction of Deep Learning. It has been demonstrated that a simple multilayer neural network architecture outperforms most state-of-the-art approaches in several NLP tasks such as named-entity recognition, semantic role labeling, and Part-of-Speech tagging. Since then, numerous complex deep learning-based algorithms have been proposed to solve NLP tasks [39].

Text classification, also known as text tagging or text categorization, is a NLP task that assigns one or more categories to a piece of text based on its features. The process of text classification involves training a machine learning or deep learning model on a labeled dataset. The labeled dataset contains examples of text along with their corresponding labels. The model learns from these examples to recognize patterns and associations between the features present in the text and the corresponding categories.

While text classification has many applications in various domains, such as sentiment analysis, spam detection, news classification, and customer feedback analysis, it is highly dependent on the availability of labeled data. Manually labeling a dataset is costly, time-consuming, and sometimes unreliable since it depends on the person who is doing the labeling.

1.2 Motivation

The *Portal da Queixa*[1] is an online platform where people can express their opinions and complaints on products, services, and brands. Currently, the website is visited by millions of Portuguese people every month and with only three steps we can post a complaint. After the complaint is validated, a notification is sent by email to the brand in question.

A typical complaint describes a problem with a product or service, a narrative on the customer's attempts to resolve the issue, and the interaction with the company.

Given the large number of complaints generated daily, a system capable of predicting the reason of the complaint could decrease customer service spending and time and increase customer satisfaction, since the customer is redirected to the right department.

In this study, we investigate the issue of automatic complaint classification in the Portuguese language. While there is extensive research on text classification and natural language processing techniques, there is a lack of research focusing on the classification of customer complaints in Portuguese, particularly in the context of delivery service companies. As such, this research aims to fill this gap in the literature.

1.3 The Main Goal

The main goal of this thesis is to address the challenge of categorizing customer complaints in the Portuguese language. In particular, what data representation techniques should be used, what classification algorithms perform the best, and how to perform feature engineering and selection to improve the algorithms' performance.

The exploratory data analysis provides insights about what distinguishes the complaints, identifies patterns, and understands the language used by customers when making specific complaints. These insights can help this particular business improve its website and customer service.

[1] https://portaldaqueixa.com/

In addition to that, we aim to analyze the output of the text classification. By studying the misclassifications of the model, it is possible to provide insights about the limitations of the models and help analysts understand the types of texts that are particularly difficult to classify and may require human input or different labels.

In summary, this study aims to address the following research questions:

1. What machine learning algorithms are most effective when applied to text classification problems?
2. How does the performance of traditional machine learning algorithms compare to deep learning models in classifying customer complaints?
3. What is the impact of using different feature encoding techniques, such as TF-IDF, word embeddings, or contextualized representations, on the accuracy of customer complaint classification?
4. Can the integration of keyword extraction using YAKE!, event extraction using text2story, and named entities improve the accuracy of customer complaint classification in Portuguese?
5. How can the findings from the exploratory analysis and text classification guide the development of more accurate labels for customer complaints, considering the identified patterns or themes in the data?

1.4 Methodology

The research methodology employed in this study follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. The research will involve several stages, including data understanding and preprocessing, feature engineering, model development and evaluation, and the interpretation of results. NLP techniques such as text preprocessing, word embeddings, and supervised machine learning algorithms will be utilized to achieve the research objectives.

1.5 Contributions

The contributions of this thesis can be summarized as follows:

1. Curation of a Portuguese dataset for text classification tasks of customer complaints. The dataset was obtained from a website using an API, where the data was already labeled. The dataset went through preprocessing and cleaning operations.
2. Comparative analysis of different classification approaches, providing a benchmark for future research and industry practices.

1.6 Organization

The remainder of this document consists of 8 sections that are structured as follows.

- **Chapter 2** presents some theoretical background needed to develop the research.
- **Chapter 3** reviews the existing literature on customer complaint data classification. It discusses the approaches and techniques that are commonly used in this kind of problem.
- **Chapter 4** describes the CRISP-DM methodology, explaining its stages and tasks. It discusses the data understanding process, data preprocessing steps, feature extraction techniques, modeling, and evaluation reasoning.
- **Chapter 5** presents the experiments and their results obtained by following the methodology defined in the previous chapter. The results are interpreted and discussed concerning the research objectives.
- **Chapter 6** summarizes the main findings of the study and addresses their implications for the field of customer complaint categorization and understanding. It also mentions the limitations of our research and areas for future improvement.
- **Chapter 7** and **8** contain more detailed information about the results of our experiments, and some examples of BERT's predictions analysis using SHAP, respectively.

Chapter 2

Background

This chapter provides essential information and theoretical explanations of models and techniques employed in this research. These concepts are fundamental to understanding our analyses and findings.

2.1 Text Encoding

Text encoding in machine learning refers to the process of converting textual data into a numerical representation that can be understood and processed by machine learning algorithms. Some common text encoding techniques are Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word Embeddings.

2.1.1 Bag of Words

The Bag of Words model (BoW) represents a document as a vector, where each column corresponds to a word or term, and the cell values correspond to the frequency of each word in the document. BoW does not consider the order of words in a document, nor the semantic relationship between words [20].

2.1.2 Term Frequency and Term Frequency-Inverse Document Frequency

Term frequency (TF) calculates the frequency of a word in a document. Inverse Document Frequency (IDF) gives a higher weight to words that occur only in a few documents. The reasoning behind this is that terms that are limited to a few documents are useful for discriminating those documents from the rest. Term Frequency-Inverse Document Frequency (TF-IDF) combines TF and IDF by multiplying them as shown in Equation (2.1) [20]:

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}) \tag{2.1}$$

where N is the number of documents, d is the document, and $df(t)$ is the number of documents containing the term t in the corpus. The fewer documents a term occurs, the higher the logarithm. The lowest weight of 0 is assigned to terms that occur in all the documents.

After applying TF-IDF, texts can be represented as a vector of dimension equal to the number of words in the vocabulary. The value corresponding to each word represents the importance of that word in a particular document.

Although TF-IDF tries to overcome the problem of common terms in the document, it does not take into consideration the similarity between the words in the document and the order of words in the phrase.

2.1.3 Word2Vec

More complex language models have been developed in recent years, such as word embeddings. Embeddings are short dense vectors, usually with dimensions between 50-1000 that can incorporate concepts such as similarity of words and part-of-speech tagging [17]. Word embeddings represent words in a vector space where words with similar meanings are located closer together. Moreover, they incorporate contextual information, i.e., words with different meanings will have distinct vectors depending on their context in a sentence. The most used word embedding methods are Word2Vec and GloVe.

Word2Vec models were proposed by Mikolov et al. [26]. They use neural network architectures, specifically the Continuous Bag of Words (CBOW) and Skip-gram models, to learn word embeddings. In both architectures, Word2Vec considers each word and a sliding window of words surrounding that word as it scans over the entire corpus. The CBOW model tries to predict a target word from a list of its surrounding words. The Skip-Gram model takes the target word and predicts the probability of a word being a context word. The goal is to maximize this probability. We do not actually care about this prediction task; instead, we will take the learned classifier weights as the word embeddings [17, 19].

2.1.4 GloVe

Global Vectors (GloVe) model was introduced by Pennington et al. [29] and it is based on a word-word co-occurrence counts matrix, i.e. for each “word”, it counts how frequently one word appears in some “context” in a large corpus. The idea behind GloVe is that the ratios of co-occurrence probabilities can encode meaningful semantic relationships between words, i.e., words that are synonymous tend to have similar ratios of co-occurrence probabilities with other words. Therefore, the goal is to produce dense vector representations for words that capture meaningful semantic relationships between words, i.e., words that are often used in similar contexts should have similar vector representations. The objective function is to learn word vectors such that their dot product equals the logarithm of the words’ probability of co-occurrence.

The algorithm then optimizes the word vectors to minimize the difference between the predicted co-occurrence values and the actual co-occurrence probabilities [29].

The advantage of GloVe is that it incorporates global statistics, which means it can capture the general semantic structure of the language.

2.2 Machine Learning

Machine Learning refers to a set of methods that can automatically detect patterns and relationships within the data, and then use the learned patterns to predict future data, or to perform other kinds of decision-making under uncertainty [27].

The three main types of ML are supervised learning, unsupervised learning, and reinforcement learning [27]. This research focuses on supervised learning, which means that the model is trained on labeled data, where each data point is associated with a known target value. If the target is a continuous numerical value, then the task is regression. If the target value is categorical or discrete, then the task is classification.

On the other hand, Deep Learning is a subset of machine learning that uses neural networks with multiple layers to analyze complex patterns and relationships within the data. It is inspired by the structure and function of the human brain. Deep learning algorithms are more accurate than machine learning algorithms, but they are also more difficult to interpret.

2.2.1 Classical Machine Learning Techniques

Classical machine learning techniques refer to the techniques in the field of machine learning that have been developed and used before the rise of deep learning. They sometimes provide a baseline for comparison when exploring more advanced approaches like deep learning techniques. Some common classical machine learning techniques include Linear Regression, Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), Naive Bayes, k-nearest Neighbors, and others. We explain the ones that we devoted more attention to throughout our research.

XGBoost (eXtreme Gradient Boosting) is an ensemble learning method that uses a sequence of weak learners, typically decision trees, to create a strong learner. During the training process, XGBoost uses a technique called gradient boosting, where the algorithm calculates the gradients of the loss function with respect to the predictions made by the previous decision trees and then fits a new tree to the gradients. XGBoost has more capabilities compared to traditional Gradient Boosting like the use of regularization techniques and parallel processing [8, 16]. XGBoost provides a measure of feature importance, which is helpful in feature selection, understanding the data, and understanding the model's behavior.

Support Vector Machine (SVM) works by finding an optimal decision boundary, called a hyperplane, that separates different classes in the feature space. The goal is to maximize the margin, which is the distance between the hyperplane and the closest data points from each class. The data points that are closest to the hyperplane are called support vectors [1]. If the data is not linearly separable in the original feature space, SVM uses a technique called the “kernel trick” that maps the input features into a higher-dimensional space where the classes become linearly separable [1].

2.2.2 Deep Learning - Transformers

Transformers are a type of deep learning model architecture that has gained popularity in recent years, especially in NLP tasks, due to their suitability for handling sequential data, such as natural language text.

Transformers were proposed by Vaswani et al. [38]. The original Transformer architecture follows an encoder-decoder structure and the novelty is the self-attention mechanism, which allows the model to capture dependencies between different words or tokens in a sequence. Unlike traditional Recurrent Neural Networks or Convolutional Neural Networks, transformers can process the entire sequence in parallel, which means that they can be more efficient to implement [17].

The input is fed to a word embedding layer to map the words to vectors. Next, positional encodings are added to the input embeddings to incorporate information about the position of the tokens in the sequence. Then, this information is passed to the first encoder [38].

The encoder is composed of a stack of six identical layers. Each layer has a multi-head self-attention mechanism and a fully connected feed-forward network. The authors employ a residual connection around each of these layers, followed by layer normalization [38]. The job of the encoders is to produce an abstract representation of the input sequence that captures the relationships between words.

The decoder is also composed of a stack of six identical layers. The first two layers are the same as the encoder layers, and the third is multi-head attention over the output of the encoder stack [38]. During the training phase of sentence translation, the output (translated) sentence is fed to the decoder. Next, the self-attention block generates attention vectors for every word in the sentence to learn their relationships. These attention vectors and the vectors from the encoder are passed into another attention block (encoder-decoder attention block) that determines how the words from the original and translated languages are mapped to each other. In summary, the decoder’s job is to generate text sequences.

The self-attention mechanism determines how much focus should be placed on other parts of the input sentence as we encode a word at a certain position. Multihead self-attention layers are sets of self-attention layers, called heads, that reside in parallel layers at the same depth in a model, each with its own set of parameters. These distinct sets of parameters allow each head to learn different kinds of relationships between words [17].

Bidirectional Encoder Representations from Transformers (BERT) was first introduced by Devlin et al. [10] and it is a multi-layer bidirectional Transformer encoder based on the original implementation of Transformers [38]. Since BERT's goal is to generate a language model, only the encoder mechanism of the Transformer architecture is necessary. The BERT (base) model has 12 stacked Encoder blocks, 768 hidden units, and 12 attention heads. BERT (large) has 24 stacked Encoder blocks, 1024 hidden units, and 16 attention heads [10].

BERT is trained using two unsupervised tasks: Masked Language Model and Next Sentence Prediction. The Masked Language Model randomly masks some of the tokens from the input, and the goal is to predict the masked word based on the words on its right and left sides. The Next Sentence Prediction task takes pairs of sentences and learns to predict if the second sentence is the next sentence in the original text [10].

For finetuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks (question answering, classification, NER, etc.). Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters [10].

2.3 Some Natural Language Processing Tasks

There are many tasks studied in this field, such as Named Entity Recognition, Part-of-Speech Tagging, Text Summarization, Text Classification, Sentiment Analysis, Topic Modeling, Question Answering, among others. Throughout this work, we will mention Text Classification, NER, Keyword Extraction and Event Extraction, so these are the tasks that we will detail in this section.

Text Classification involves assigning predefined categories or labels to texts based on their content. Text classification is widely used in various applications like sentiment analysis, spam detection, topic classification, document classification, and others.

Named-Entity Recognition (NER) is a technique in NLP that involves identifying and classifying named entities. A named entity is an object, concept, or entity that can be referred to by a proper name: a person, a location, or an organization. The four most common entity tags are: PER (person), LOC (location), ORG (organization), or GPE (geo-political entity). However, the term named entity is commonly extended to include other things like dates and times [17].

Keyword Extraction is a technique in NLP that involves identifying the most important words or phrases in a text. The keywords can be helpful for various tasks like summarization, information retrieval, topic modeling, and document categorization. Notably, YAKE! is an unsupervised approach that relies on statistical text features in a document to select the most important keywords. It works on texts of different languages, sizes, and domains, as it does not need to be trained on a particular set of documents, dictionaries, or corpora [6].

Event Extraction is a task in NLP that focuses on identifying specific events or occurrences in a text. An event refers to a specific occurrence that happened in a specific point, or interval, in time. Most event mentions correspond to verbs, and most verbs introduce events, but that is not always the case [17].

2.4 Model Interpretability and Explainability

Model explainability involves providing insights into why a particular prediction was made by the model. Being able to correctly interpret a prediction model's output is extremely important, as users are more likely to trust and accept the model's output. Moreover, we can identify errors and biases in the training data more easily, and understand the process being modeled [24]. Examples of explainability techniques include feature importance scores, partial dependence plots, SHAP[1] (SHapley Additive exPlanations) values, and LIME [2] (Local Interpretable Model-agnostic Explanations).

On the other hand, interpretability involves creating models that are comprehensible by humans. Techniques for model interpretability include using simpler models like decision trees or linear regression, and rule-based models [25].

SHAP, introduced by Lundberg and Lee [24], is based on game theory, specifically the concept of Shapley values. Shapley values are a way to fairly distribute the "contribution" of each player in a cooperative game. In the context of machine learning, the "players" are the input features, and the "game" is the predictive model. SHAP is model-agnostic, meaning it can be applied to a wide range of machine learning models, without requiring modifications of the model itself. Moreover, it can provide both local and global interpretability by summarizing Shapley values across multiple predictions [24].

2.5 CRISP-DM (Cross-Industry Standard Process for Data Mining)

CRISP-DM (Cross-Industry Standard Process for Data Mining) is a methodology that is used for conducting data mining and analytics projects. Saltz [31] describes the main steps of this methodology and the tasks that need to be done in each one of them:

1. **Business Understanding**: The understanding of the project objectives, requirements, and constraints from a business perspective. It focuses on identifying the problem and the project goals.

2. **Data Understanding**: The collecting and exploring of the data to gain insights into its quality and relevance to the business problem. This phase also helps identify the steps

[1] https://github.com/shap/shap

[2] https://github.com/marcotcr/lime

needed for data preprocessing. Exploratory data analysis is often performed to discover patterns and relationships in the data.

3. **Data Preparation**: The cleaning, transforming, and preprocessing of the data to make it suitable for analysis. Tasks may include handling missing values, dealing with outliers, selecting relevant variables, and extracting new features.

4. **Modeling**: The building and applying data mining models to address the project objectives.

5. **Evaluation**: To evaluate the models created in the previous step. Their performance is measured using appropriate evaluation metrics. At this point, we determine if the models meet the business requirements or if we need to go back a few steps.

6. **Deployment**: It includes a deployment plan, a monitoring and maintenance plan, a final report, and an overall project review.

CRISP-DM provides an organized framework that can be adapted to different project domains. It guides data scientists through the entire data mining process, making it easier to handle.

Chapter 3

Related Work

This chapter provides a comprehensive review of the existing literature and research that is aligned with the topic of text classification, specifically focusing on the classification of customer complaints. The purpose of this chapter is to provide a context for the current study by examining similar works, methodologies, challenges, and solutions adopted by other researchers in the field of customer complaints analysis. We divide this section into the classification of customer complaints in English or other languages, the classification of customer complaints in Portuguese, techniques to handle imbalanced datasets and techniques to explain and interpret ML models. Finally, the last section summarizes the described works and identifies the research gap that this study aims to address.

3.1 Customer Complaint Classification

Elfardy et al. [12] trained classifiers on a four-language (English, Spanish, French, and Japanese), multi-label corpus released as part of the shared task on *"Customer Feedback Analysis"* at IJCNLP 2017. The task was to classify each piece of feedback into one or more categories like "bug", "comment", "complaint", "meaningless", "request", or "undetermined". They increased the dataset's size by translating each feedback to the other available languages. The authors faced challenges due to short feedback length (around 10-13 words) and imbalanced label distribution. In our research, we faced the second challenge of the imbalanced dataset as well, with two labels being more common.

The authors started by tokenizing and stemming the English and Spanish data using the Stanford Core NLP toolkit. For French, they used an in-house tool for both tokenization and stemming while for Japanese, they used MeCab to tokenize the data. Then, they used standard n-gram features of size 1 to 3, and pre-trained FastText word embeddings to train a binary Random Forest classifier. To train a Logistic Regression, they experimented with different feature sets for each language, including TF-IDF score of n-grams of size 1 to 3 that appear in more than five training instances, stemmed N-grams, and the average of embeddings of all words in a given instance. For English, Spanish, and French languages, the best results were achieved by combining n-grams with word embeddings. The difference among these three languages was in whether the optimal setup involved using basic n-grams, stemmed n-grams, or both. However, for Japanese, adding word embeddings decreased performance. This might be due to differences in how the text was tokenized for training the embeddings compared to this model's tokenization. Finally, they used pre-trained Facebook's FastText embeddings for each language.

Following that, the authors used three methods to train their classifiers: Logistic Regression with L2 regularization and the inverse of regularization strength (C) of 100, Random Forest, and LSTM. The performance of the models was evaluated according to Micro and Macro-Average F1-Score because the dataset was highly imbalanced. The LSTM architecture included 80 nodes in most cases, except for French, where each LSTM has 100 nodes, to improve performance. Dropout was applied with a rate of 0.1 on both the input and hidden layer weights. The last layer was a fully connected layer with six output nodes corresponding to the six output classes. The optimizer used was RMSProp optimizer. The authors trained a separate LSTM for each language, for 10 epochs with a learning rate of 0.001 and optimized using a categorical cross-entropy loss function. Overall, they achieved the worst results with Random Forest for all languages. LSTM performed best in French and Japanese languages, achieving a Macro Average F1-score of 48.7 and 53.8 on the test set, respectively. Logistic Regression performed best in English and Spanish languages, achieving a Macro Average F1-score of 48.8 and 56.0 on the test set, respectively. All three models performed best on the most dominant classes – "complaint" and "comment", and performed worse on "meaningless" and "undetermined" classes because there were not enough training examples for the model to be able to distinguish these classes.

Similarly, Silva et al. [32] propose a method to automatically categorize IT service incident tickets in English, made by customers to a company. The dataset consists of 10,000 incident tickets assigned to one of the 10 following categories: application, collaboration, enterprise resource planning (ERP), hosting services, network, security and access, output management, software, workplace, and support. Alongside the description and label, each incident includes caller id, severity, and contact source information. The dataset is balanced with 1,000 instances per class, as opposed to our highly imbalanced dataset.

The authors preprocess the complaint description using techniques like tokenization, stopword removal, and stemming. Additonally, they transformed the preprocessed complaint texts into a feature vector using TF-IDF. They examined three feature combinations for each algorithm: using caller id, severity, and contact source; using only incident description encoded using TF-IDF; and utilizing all attributes. The authors were trying to assess the impact of using the description on categorization and therefore, identify the best features for optimal performance. They concluded that using only the incident description leads to higher accuracy when compared to using only numerical attributes (caller id, severity, and contact source). Moreover, when they combined all attributes, including the incident description, the accuracy further improved.

To automate the classification, they used SVM and KNN, with SVM consistently outperforming KNN, particularly in cases involving textual data. When using only textual data, KNN achieved an accuracy of 80%, while SVM achieved an accuracy of 86%. When using a combination of numerical and textual data, KNN achieved an accuracy of 82%, while SVM achieved an accuracy of 89%.

Table 3.1: Summary of Customer Complaint Classification Studies.

Study	**Year**	**Text Preprocessing Techniques**	**Text Encoding Techniques**	**Classification Models**	**Best Results**
Elfardy et al. [12]	2017	Tokenization and stemming	N-gram features (size 1 to 3), FastText word embeddings, TF-IDF, stemmed N-grams	Logistic Regression, Random Forest, LSTM	Macro-Average F-score between 48.7% and 56.0% with LSTM and LR
Silva et al. [32]	2018	Tokenization, stopword removal, and stemming	TF-IDF	SVM, KNN	Accuracy of 89% with SVM

3.2 Portuguese Customer Complaint Classification

Regarding the Portuguese language, most of the research found on user-generated data is used for sentiment analysis [3, 7, 9, 11, 34, 35] and offensive language detection [2]. To the best of our knowledge, few works investigate the classification of customer complaints in the Portuguese language.

Based on our research, the work that faces similar challenges and works with a dataset similar to ours is conducted by Lopes-Cardoso et al. [23]. For this reason, we are going to extensively explain their work and compare it to ours. In this paper, the authors consider a dataset of complaints that reached the Portuguese Economic and Food Safety Authority (ASAE), via email or their website. The dataset consists of 150,669 samples, and each complaint has, on average, 255 tokens. Each complaint was manually classified by an officer, according to three key dimensions: economic activity, infraction severity, and competent authority to handle a complaint. The economic activity has eleven categories - Primary production, Industry, Restauration, Wholesalers, Retail, Direct selling, Distance selling, Production & trade, Safety & environment, Service providers, and No activity identified - and each one of them has subclasses. The second key dimension, infraction severity, falls into one or more classes - crimes, administrative infringements, and simple consumer conflicts. The third key dimension is competence, which determines which entity should address the problem - ASAE or others.

Another similar task was conducted by Caldeira et al. [5]. The research aimed to classify summarized complaints received by another Portuguese public institute. Their complaints dataset consisted of 4,459 complaints spread over 17 different labels.

The next sections explore these two works in more detail, including some of the challenges they faced and the techniques they employed.

3.2.1 Main Challenges and Solutions

Lopes-Cardoso et al. [23] faced some challenges due to the imbalanced data and difficulty in distinguishing classes boundaries. Accurately determining the economic activity was a challenging task due to cases where an operator might be involved in multiple activities, and sometimes the complaint text itself did not explicitly state the targeted operator. The top 3 classes of economic activity (Restauration, Service providers, and No activity identified) together represent 72.58% of the data, while the smallest class contains only 0.02% of the data. Additionally, the third most representative class (No activity identified) is one where the human was not able to identify an appropriate economic activity. Initially, classifying the second key dimension, was a multi-label problem. However, to simplify the task and deal with the imbalance in data distribution, the authors focused on predicting the highest severity infraction present in the list of infractions associated with a given complaint. Even with this change, the dataset was still imbalanced, with the most relevant class (crime) being the least represented, with approximately 5% of the data. Finally, since the third key dimension could involve more than one competent authority, the authors transformed this into a binary classification task: whether ASAE should be involved in analyzing the complaint or not (together with other institutions, or not). 63% of the complaints fell into ASAE "jurisdiction".

In summary, Lopes-Cardoso et al. [23] faced problems such as extremely imbalanced hierarchical multi-class tasks, sparse multilabel tasks, and fuzziness of the label boundaries. To tackle this problem, they explored several approaches, including different preprocessing techniques, tokenization strategies (including subword-based approaches), sparse (feature-based) and dense (word embeddings) representations, and language models.

Caldeira et al. [5] faced challenges such as small training datasets and imbalanced class distribution. To handle the label imbalance issue, they analyzed a multistage classification method: in the first stage they considered the top 3 classes (almost 80% of the data) and a new label "others" that included all the remaining minority labels; in the second stage they considered the remaining classes that were not part of the top 3 (20% of the data), and they grouped in class "others" all the categories with less than 15 examples; in the full data stage, they considered the top 8 classes along with the remaining classes grouped under the label "others". Additionally, the authors translated complaints with low representation into multiple languages and back to Portuguese to create new artificial examples.

3.2.2 Methods

Lopes-Cardoso et al. [23] performed different cleanup approaches, including stripping all HTML code from complaints received via email and removing headers, URLs, and email addresses.

As preprocessing steps, they tried different combinations of the following techniques: baseline techniques, spell checking, synonym substitution, removal of accentuation, numerical data removal (including dates/times), and stemming. The baseline techniques included tokenization and lemmatization using Stanza, followed by lowercasing tokens and removing punctuation marks and stopwords. The goal was to reduce the size of the vocabulary. Unfortunately, the authors found that their results from these experiments did not lead to clear conclusions regarding the best preprocessing techniques in terms of both model performance and feature analysis. However, they observed that a combination of spell-checking and removing numerical data appeared to be a reasonable approach to creating a model suitable for real-world use.

They also tried subword tokenization, in conjunction with the baseline techniques, excluding lemmatization, and replacing the Stanza tokenizer with one of the following: WordPiece tokenizer, based on the BERT model "bertbase-multilingual-cased", and Cross-Lingual Language Model (XLM) tokenizer, based on the "xlm-mlm-100-1280" pre-trained model. Both models were available in HuggingFace's Transformers library and were trained on multilingual resources (one of them being Portuguese language). This experiment targeted only the economic activity prediction task. They used SVM and observed a drop in performance as compared to the base techniques, from F1-Macro of 0.5640 (Stanza) to 0.5241 (BertTokenizer) and 0.5217 (XLMTokenizer).

As feature extraction techniques, they used 1-grams and TF-IDF for feature representation. The use of larger n-grams ($n > 1$) did not show any improvement. Additionally, they used pre-trained Word2Vec embeddings [15], FastText model embeddings, and the BERT "bert-base-multilingual-cased" model embeddings provided by the HuggingFace library.

Following that, they trained the follwing classifiers: SVM with a linear kernel; Stochastic Gradient Descent (SGD), and Random Forest. For reasons of explainability, they decided not to use neural networks. The SVM classifier was configured to use the L2 norm as a penalty, squared hinge loss, and the one-vs-rest strategy for multiclass. The SGD classifier was configured to use the L2 norm as a penalty and the smoothed Hinge Loss. The Random Forest classifier was configured to use 100 estimators. They used it both as a multiclass classifier and in a one-vs-rest strategy, to compare with SVM. SVM obtained the best overall results, for all three classification tasks, achieving a Macro F1-Score of 0.5640, 0.6542, and 0.7748 for the economic activity prediction task, infraction severity prediction task, and competence prediction task, respectively.

Furthermore, they tried three more approaches on the economic activity prediction task: pre-trained Word2Vec embeddings with a Convolutional Neural Network, the FastText model, and the pre-trained BERT model "bert-base-multilingual-cased", with one fully-connected hidden layer with 768 neurons on top, and a final softmax layer with 11 neurons (the number of classes). All BERT layers were fine-tuned for their task, for 10 epochs, with a batch size of 8, the initial learning rate of 4×10^{-6} that was increased during the first epoch, reaching 1×10^{-4}, and then reduced for 9 epochs. The Macro F1-Score results were 0.4669 for Word2Vec+CNN, 0.5119 for the FastText model, and 0.5718 for BERT. The models that are based on word embeddings performed worse, and the BERT-based model was able to obtain a small improvement in both accuracy and macro-F1, lower than 1%.

Similarly, Caldeira et al. [5] preprocessed the complaints by employing lowercasing, removing numerical data, removing special characters, removing diacritics, and expanding acronyms. In this case, since the tokens were already part of a short corpus, the additional processing only led to marginal improvements. Then, the authors used the TF-IDF model to get the feature vectors for each complaint. For the BERT-based approach, instead of TD-IDF, the tokens were preprocessed using the model encodings. Several classification models were experimented with, including more traditional approaches like Naive Bayes, KNN, SVM, and XGBoost models, and deep learning-based approaches like Multilingual-BERT. The study was divided into three stages:

- For the first stage experiment (top 3 classes and a new label “others” with the remaining minority labels), when using the original dataset, Naive Bayes achieved an F1-Score of 0.91. However, when working with the augmented dataset, SVM achieved a higher F1-Score of 0.94.
- For the second stage experiment (classes that were not part of the top 3, and a new class “others” with all the categories with less than 15 examples), when using the original dataset, SVM achieved an F1-Score of 0.25. However, with the augmented dataset, Multilingual-BERT achieved a significantly improved F1-Score of 0.59.
- For the full data stage experiment (top 8 classes along with the remaining classes grouped under the label “others”), when using the original dataset, SVM achieved an F1-Score of 0.85. Interestingly, with the augmented dataset, SVM achieved an even higher F1-Score of 0.90, demonstrating the benefits of data augmentation in this context.

Table 3.2: Summary of Portuguese Customer Complaint Classification Studies.

Study	**Main Challenges**	**Preprocessing Techniques**	**Text Encoding**	**Models Techniques**
Lopes-Cardoso et al. [23] (2021)	Imbalanced hierarchical multi-class tasks, sparse multilabel tasks, and fuzziness of the label boundaries	Cleanup of HTML code, headers, URLs, email addresses; tokenization, subword tokenization, lowercasing, stemming/lemmatization, spell checking, synonym substitution, removal of accentuation, numerical data, punctuation, stopwords, dates/times	1-grams and TF-IDF, pre-trained Word2Vec embeddings, FastText embeddings, Multilingual BERT embeddings	SVM with linear kernel, SGD, Random Forest, CNN, FastText model, Multilingual BERT. Highest Macro F1-Score of 0.7748 with SVM
Caldeira et al. [5] (2022)	Small training datasets, imbalanced class distribution	Lowercasing, numerical data removal, special character removal, diacritic removal, and acronym expansion	TF-IDF, Multilingual BERT embeddings	Naive Bayes, KNN, SVM, Multilingual BERT, XGBoosting. Highest F1-Score of 0.94 with SVM and augmented dataset

3.3 Methods for Multiclass Imbalanced Data Classification

When one class has significantly more examples than the others in a dataset, we have an imbalanced dataset. Models trained on imbalanced datasets tend to become biased towards the majority class, often leading to poor performance over the minority classes. To address class imbalance issues, several techniques have been proposed, such as data-level approaches that change the dataset to balance the class distribution before feeding it to the algorithm (random over-sampling, random under-sampling, synthetic minority over-sampling technique (SMOTE), direct oversampling and data augmentation), and algorithm-level approaches that modify the algorithm itself to make it more suitable for imbalanced datasets (modifying the loss function, considering different misclassification costs for each class or ensemble methods) [37].

Moreover, many evaluation metrics have been proposed to measure the classifier performance when dealing with imbalanced data problems such as Precision, Recall, F-measure, AUC, G-mean, Kappa, and MCC [37].

Glazkova [14] addresses the class imbalance problem in multiclass topic classification for text fragments containing biographical information using different supervised learning methods, including KNN, SVM, Feed-Forward Neural Network (FNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (Bi-LSTM). To feed the information to the algorithms, the author experimented with different text representations, including Bag-of-Words, Bag-of-Words + TF-IDF, and Word2Vec. To mitigate the impact of class imbalance on classification results, the author compares different oversampling techniques, including Synthetic Minority Oversampling Technique (SMOTE), Borderline SMOTE, Adaptive Synthetic Sampling (ADASYN), and random oversampling. Glazkova [14] concludes that the oversampling methods improve text classification accuracy and F1-macro score, particularly for KNN and SVM methods. On the other hand, Neural Networks generally show less sensitivity to oversampling techniques. In most cases, the best results were achieved using the ADASYN algorithm and the random oversampling technique. About SVM, Borderline SMOTE and ADASYN appear to be consistently good choices for improving SVM performance when combined with the TF-IDF representation. For the SVM + Word2Vec configuration, SMOTE and ADASYN seem to consistently improve SVM's performance.

Rathpisey and Adji [30] study the application of four resampling methods – Random Oversampling (ROS), SMOTE, ADASYN, and Random Undersampling (RUS) – to address class distribution imbalance in a hate speech dataset. The study employs three machine learning classifiers: SVM, Logistic Regression, and Naive Bayes. Experimental results demonstrate that oversampling techniques (ROS, SMOTE, ADASYN) lead to improved performance in all classifiers while undersampling only benefits the Naive Bayes classifier. Among the methods and classifiers, Logistic Regression with ROS achieves the best outcomes, with a 91% accuracy and F1-Score of 0.95.

Handling imbalanced datasets is still an open research problem, especially when dealing with multiple classes. Current solutions mostly focus on cases where there are two classes, where the majority and minority class boundary is well defined. However, these methods do not always work well for imbalanced scenarios with more than two classes. Text classification is also a research area that has very few works devoted to handling the class-imbalance. [36]

Table 3.3: Summary of Methods for Multiclass Imbalanced Data Classification Studies.

Study	**Year**	**Objective and Methods**
Glazkova [14]	2020	Tackle class imbalance in multiclass topic classification for text fragments using supervised learning methods with different text representations and oversampling techniques (SMOTE, Borderline SMOTE, ADASYN, random oversampling). Oversampling methods, especially ADASYN and random oversampling, improve text classification accuracy and F1-macro score for KNN and SVM. Neural Networks show less sensitivity to oversampling.
Rathpisey and Adji [30]	2019	Study the application of resampling methods (ROS, SMOTE, ADASYN, RUS) in a hate speech dataset with three classifiers (SVM, Logistic Regression, Naive Bayes). Oversampling techniques (ROS, SMOTE, ADASYN) improve performance in all classifiers while undersampling benefits Naive Bayes. Logistic Regression with ROS achieves the best outcomes.

3.4 Model Interpretability and Explainability Techniques

Lai et al. [21] compare built-in feature importance from traditional models such as linear SVM and neural models with attention mechanisms, as well as post-hoc importance based on LIME and SHAP. They find out that features considered important by deep learning models (like LSTM and BERT) are distinct from those identified by traditional models (SVM and XGBoost). When post-hoc methods (like LIME) are applied to different models, they tend to make the important features identified by these models more similar to each other, compared to the built-in methods (such as attention mechanisms in deep learning models). Even when the analysis focuses specifically on the top features, different approaches may still yield diverse outcomes. Moreover, even when two models agree on the predicted label for a given instance, the features they identify as important may not necessarily be very similar or consistent. For these reasons, we will be trying not only SHAP but also built-in methods, if available.

Occhipinti et al. [28] study model interpretability and feature contributions in the context of email spam classification. The authors employ the SHAP method to quantify the impact of different features on the classification outcomes of their machine learning models (Random Forest, XGBoost, and a Message Passing Neural Network).

Bangerter et al. [4] address two subtasks in the SemEval-2023 Task 3, which is focused on *"Detecting the Genre, the Framing, and the Persuasion techniques in online news in a multilingual setup"*. The first task is to classify news articles into one of three categories: opinion, report, or satire, and the second task is to identify and reveal the persuasion techniques used in each paragraph of news articles, out of 23 defined methods. On the first task, the authors use SHAP to get a list of tokens that are inducing the model into misclassifications. These tokens were filtered out from the texts during the preprocessing step. Following this, the model was retrained, leading to improved performance. In the second task, SHAP is used to construct vocabularies of essential words that characterize each persuasion technique. Words in the input text that closely match words in the SHAP vocabulary are included in the input paragraph, which is used for multiclass classification.

Table 3.4: Summary of Model Interpretability and Explainability Techniques Studies.

Study	**Year**	**Objective and Methods**
Lai et al. [21]	2019	Compare feature importance from traditional models (e.g., linear SVM) and neural models with attention mechanisms, along with post-hoc importance using LIME and SHAP. They find out that features considered important by deep learning models differ from traditional models. Features considered important by Post-hoc methods tend to be similar even when applied to different models. Models may agree on predictions but identify different important features.
Occhipinti et al. [28]	2022	Study model interpretability and feature contributions in email spam classification using SHAP. Employ SHAP to quantify feature impact on the classification outcome.
Bangerter et al. [4]	2023	Classify news articles and persuasion techniques. Use SHAP to identify tokens inducing misclassifications and filter them out. Use SHAP to construct vocabularies of essential words for each class.

3.5 Summary and Research Gap

Similarly to Caldeira et al. [5] and Lopes-Cardoso et al. [23], our initial dataset was highly imbalanced. To handle this issue, we followed a similar approach of Caldeira et al. [5] of considering only the top classes. However, we chose not to combine the remaining minority classes, because we wanted to be able to provide some insights about the issues of the complaints and we would have lost a lot of information by doing so. Despite this effort, we still ended with an imbalanced dataset, with two predominant classes.

Furthermore, we encountered a similar issue with fuzzy label boundaries as described by Lopes-Cardoso et al. [23]. Despite our texts being labeled with just one class, upon further analysis, it became apparent to us that some complaints included different types of issues, and not only the one initially assigned to them. This lack of certainty in the label assignment can introduce noise into the training data, affecting the classifier's ability to generalize well to unseen instances. To address this issue, we adopted a similar approach to Lopes-Cardoso et al. [23], experimenting with various feature representations, language models, feature engineering, and feature selection.

In summary, the study conducted by Lopes-Cardoso et al. [23] and Caldeira et al. [5] closely aligns with our methodology, data type, data distribution, and challenges. We decided to follow similar preprocessing techniques, feature encodings, and classification models. Our work aims to reproduce their results to a different dataset and add more techniques to deal with class imbalance like resampling strategies, hyperparameter tuning, and feature selection. We also intend to apply techniques for explaining the predictions of machine learning models, mostly SHAP, since it can provide both local and global interpretability and different visualization options.

Chapter 4

Text Classification of Customer Complaints using CRISP-DM framework

In this chapter, we present the methodology employed for the text classification of customer complaints from *Portal da Queixa* website. The research objectives of this project are (1) to evaluate the effectiveness of machine learning algorithms in the text classification of Portuguese customer complaints, (2) to compare traditional machine learning algorithms with deep learning models, (3) to assess the impact of different feature encoding techniques on classification performance, (4) to utilize exploratory analysis and classification results to provide insights into the main issues faced by customers, and the quality of the labeling that is available.

Our research design follows a case study approach, focusing on a specific company and its customer complaint data. This approach will allow us to understand the challenges faced by the company and provide possible solutions to them.

To achieve our goals, we followed the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. The CRISP-DM framework consists of six main steps: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

4.1 Business Understanding

Figure 4.1 showcases the website known as *Portal da Queixa*. This online platform allows people to express their opinions and complaints on products, services, and brands. After the complaint is validated, a notification is sent by email to the brand in question. A typical complaint describes a problem with a product or service, a narrative on the customer's attempts to resolve the issue, and the interaction with the company. Given the vast amount of data received daily, companies like these are leveraging Artificial Intelligence, specifically Natural Language Processing (NLP), to automatically understand this information. By extracting patterns and trends from the texts, such as common issues or frequently mentioned products or services, companies can improve their customer service quality.

Pesquisa antes de Comprar.

Portal da Queixa by Consumers Trust é um marketplace de reputação que te permite **pesquisar**, **reclamar** e **comparar** qualquer marca **gratuitamente**.

Figure 4.1: *Portal da Queixa* website.

4.2 Data Understanding

The data was obtained by downloading it through an API provided by *Portal da Queixa*, using a token and password also provided by them. The original data was available in JSON format and it was transformed into a single comma-separated value (CSV) file. The data concerns customer complaints towards a Portuguese postal and delivery company.

Originally, there were 34,908 complaint records. Each one of them was composed of the following features: id, date, status, private, title, reason, reason_other, reference, incident_date, description, user, and replies. Figure 4.2 depicts the distribution of token counts within the complaints (feature “description”). Most complaints have between 0 and 200 tokens, which means that the customers get to the point quickly. For instance, in Figure 4.3 we see an example of a short customer complaint related to Delivery Conditions.

For this thesis, since our focus is on data generated by the user, we are only going to use “description” and “title” features. Since the dataset already contains information about the reason for the complaint (feature “reason”), it is possible to use it as a gold standard to train classifiers. The description is a textual field where the customers expose their problem and the title summarizes the description. The reason is selected by the user from a predefined list of options represented in the y-axis of Figure 4.4. In summary, only the title, description, and reason fields were considered.

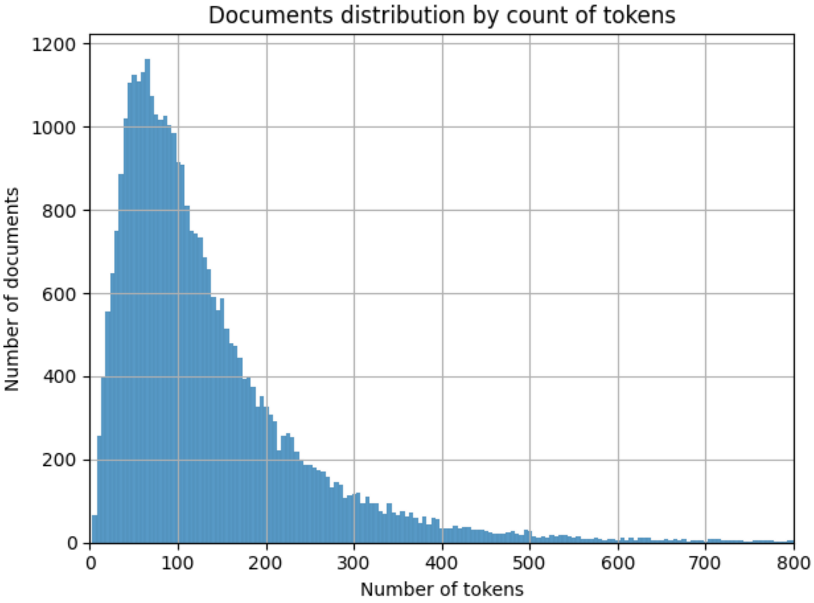
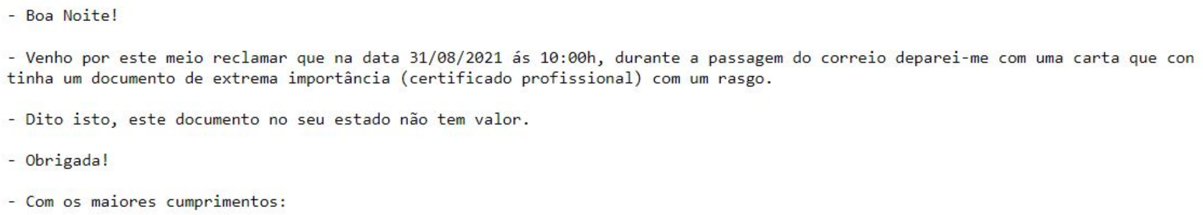


Figure 4.2: Documents distribution by count of tokens.



```
- Boa Noite!

- Venho por este meio reclamar que na data 31/08/2021 ás 10:00h, durante a passagem do correio deparei-me com uma carta que con
tinha um documento de extrema importância (certificado profissional) com um rasgo.

- Dito isto, este documento no seu estado não tem valor.

- Obrigada!

- Com os maiores cumprimentos:
```

Figure 4.3: Example of a customer complaint with label *"Condições de entrega"* (Delivery Conditions). Translation: "Good evening! I am writing to complain that on 31/08/2021 at 10:00 AM, during the delivery of the mail, I encountered a letter that contained a document of utmost importance (professional certificate) with a tear. That being said, this document, in its current state, has no value. Thank you! Best regards:".

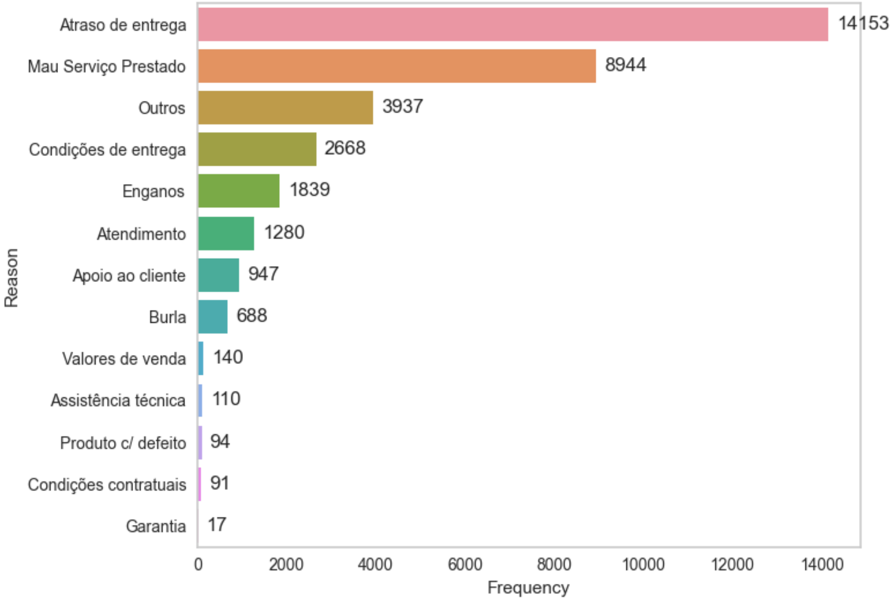


Figure 4.4: Distribution of the number of complaints per reason.

There is a large number of classes available. Figure 4.4 shows the frequencies for each class. However, we have limited data for certain classes, so the model may not have enough information to generalize well and make accurate predictions for those classes. Moreover, the classes with few observations might represent events that occur less frequently in the real world. On the other hand, classes with more samples may represent major customer problems, making them more important from the business perspective. Therefore, we decided to focus our attention on the four most common classes and train the classifiers to predict:

- *"Mau serviço prestado"*/"Poor Service Provided" (35.5% of the final dataset)
- *"Condições de entrega"*/"Delivery Conditions" (9.9% of the final dataset)
- *"Atraso de entrega"*/"Delay in Delivery" (47.8% of the final dataset)
- *"Enganos"*/"Mistakes" (6.8% of the final dataset)

We did not consider the class *"Outros"* ("Others") because it did not provide any information about the service.

4.3 Data Preparation

Data preparation is an important step in the development of machine learning models. This process involves collecting, cleaning, and transforming raw data into a structured and meaningful format that can be used and understood by machine learning algorithms. At this stage, we applied text preprocessing, text encoding, and feature extraction techniques. Text preprocessing involves cleaning the text of inconsistencies and preparing it for further processing, text encoding involves converting the data into numerical representations, and feature extraction involves selecting and transforming relevant features from the dataset.

4.3.1 Text Preprocessing

The customer complaint data went through a series of preprocessing steps, including removing irrelevant information and performing text cleaning techniques such as removing stop words and punctuation. This ensured that the data was in a suitable format for further analysis.

At this stage, a total of 206 texts written in other languages were removed, along with 6,267 texts with less than 50 tokens before preprocessing. After filtering the dataset for the four most common reasons, the dataset size was reduced to 22,430 samples.

Since we will be training text classification models using different text representation techniques such as TF-IDF, word embeddings, and BERT (as a classifier), we performed different preprocessing steps depending on the techniques that were used. The frequency-based feature engineering techniques like TF-IDF are sensitive to noise. As a result, the following steps were applied: lowercasing, transforming HTML code to Unicode, removal of newline characters and multiple spaces, removal of emails and hyperlinks, removal of numbers and punctuation, removal of stopwords using NLTK: Natural Language Toolkit[1], lemmatization using spaCy[2] and tokenization using NLTK.

In regards to the word embeddings, we used the same script that the authors of the embeddings used to clean their corpus [3]. Some of the steps applied were: emails were mapped to an EMAIL token, all numbers were mapped to 0 tokens, all URLs were mapped to a URL token, different quotes were standardized, different hyphens were standardized, HTML strings were removed, all text between brackets were removed.

BERT, a transformer-based language model, does not typically require some preprocessing steps like stemming or the removal of stopwords and punctuation. In fact, BERT is designed to use all of the information in a sentence. However, to reduce noise, some steps were applied like transforming HTML code to Unicode, removing newline characters and multiple spaces, removing emails and hyperlinks, and replacing punctuation repetition with single punctuation (e.g., “?????” was replaced by “?”).

4.3.2 Text Encoding

Text encoding is necessary to convert preprocessed texts into a numerical representation that can be understood by machine learning algorithms. To find the best TF-IDF configurations for our text classification task, we started by defining a range of possible configurations for TF-IDF. These configurations included variations in parameters such as n-gram range, maximum document frequency, and minimum document frequency. For this task, we used TfidfVectorizer [4] class provided by the scikit-learn library.

We also explored different Portuguese pre-trained Word2Vec and GloVe embeddings with a dimension of 600. The models were trained on a large Portuguese corpus with both Brazilian and European variants [15]. The numeric representation of each word is taken from Word2Vec or GloVe. Then, all the vectors are added, producing a single vector with a dimension of 600 that represents the information of the sentence. After that, we divided the sum by the number of words in the sentence. Using the mean can be useful in cases where the length of the text varies significantly. This approach may be more appropriate for tasks such as text classification or clustering, where the goal is to represent the overall meaning of a text rather than capturing specific details.

Each of these techniques represents a different approach to capturing the meaning and context of the text. We used each one of them with different text classification models. The purpose of these experiments is to understand which technique works best for our specific task and dataset.

[1] https://www.nltk.org/
[2] https://spacy.io/
[3] https://github.com/nathanshartmann/portuguese_word_embeddings/blob/master/preprocessing.py
[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

In what concerns language models feature encoding, the approach involved encoding the customer complaints using the Transformers[5] library's AutoTokenizer, adding special tokens, such as "[CLS]" (the classification token) and "[SEP]" (the separator token). To ensure uniformity in input lengths, padding, and truncation techniques were applied.

4.3.3 Feature Extraction

To extract relevant features from the complaint description, we employed various techniques such as Named Entity Recognition (NER), Keyword Extraction, and Event Extraction. These techniques enabled us to identify important entities, keywords, and events associated with each complaint. Further elaboration on the items can be found in the following item list.

- Keywords were obtained using YAKE![6]. Most of the time they contain the exact expressions related to the complaint issue because they represent the most relevant and distinctive terms in the text. However, Keywords from YAKE! were extracted based on their frequency and position in the document, and consequently may not capture all of the issues exposed in the document.
- Events were selected using Text2Story[7] package. They provide a sequence of actions leading up to the complaint. However, it may not include all relevant events due to the colloquial and diverse language present in the complaints.
- Named entities like locations and organizations were captured using a fine-tuned BERT model adapted for Named Entity Recognition (NER) tasks, that utilizes Conditional Random Fields (CRF) as the decoder[8]. Named entities provide additional context about the complaints, e.g., if many complaints are coming from a particular region. However, they may not be accurate in all cases.

To maintain consistency, we applied the same preprocessing steps that were used for the baseline features before incorporating these new additions into our models. The preprocessing techniques were dependent on the feature encoding method used.

It is important to note that "entities" had 5,091 missing values, and "events" had 23. For the feature "events", the number of records with missing values is relatively small compared to the overall dataset, so we could consider dropping those rows. However, this would result in datasets with different sizes, making direct comparisons challenging. Therefore, in order to keep all the rows and ensure that all datasets have the same size, we filled the missing values with an empty string and combined all the features in one string.

[5] https://huggingface.co/docs/transformers/index
[6] http://yake.inesctec.pt/
[7] https://pypi.org/project/text2story/
[8] https://huggingface.co/arubenruben/NER-PT-BERT-CRF-HAREM-Selective

4.4 Modeling

This step involves applying various machine learning algorithms and techniques to create models that can make predictions, classify data, or discover patterns. It encompasses model selection, model training, hyperparameter tuning and other optimization strategies, and model evaluation and comparison.

4.4.1 Model Selection

We considered various text classification models to address our task. The models were chosen based on a review of existing literature in a similar domain.

The research suggested that the most used models for text classification tasks, particularly in the context of customer data, were Support Vector Machines, Decision Trees, Random Forests, and K-nearest neighbors. Moreover, we included Naive Bayes in our model selection for its simplicity and ability to handle high-dimensional data. Regarding the implementation of these, the models were trained using the scikit-learn library[9]. We trained Linear Support Vector Classification (Linear SVC)[10], which is similar to SVC[11] with parameter kernel set to "linear", but implemented in terms of liblinear [13], so it has more flexibility in the choice of penalties and loss functions and tends to be faster to converge the larger the number of samples is.

Additionally, we included XGBoost (Extreme Gradient Boosting) and Gradient Boosting because they have gained significant popularity due to their performance and ability to handle complex relationships in the data. They also provide insights into feature importance, that can help us understand which aspects of the complaints contribute significantly to their respective classes. Gradient Boosting was trained using the scikit-learn library, and XGBoost was trained on XGBoost library[12].

Lastly, we decided to use a pre-trained language model in our experiments because they are capable of capturing contextual information, handling long-range dependencies, and understanding language patterns. Moreover, previous research in similar NLP tasks demonstrates the effectiveness of BERT. The language model chosen was BERTimbau, introduced by Souza et al. [33]. The model was initialized with multilingual BERT (mBERT) and further trained on brWaC corpus - Brazilian Web as Corpus, a crawl of Brazilian webpages that contains 2.68 billion tokens from 3.53 million documents [33]. Due to computational reasons and the size of our dataset, we decided to apply BERTimbau Base[13], which features 12 layers with a hidden size of 768, 12 attention heads, and a total of 110 million parameters [33]. BERT was trained

[9] https://scikit-learn.org/stable/

[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[11] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[12] https://xgboost.readthedocs.io/en/stable/

[13] https://huggingface.co/neuralmind/bert-base-portuguese-cased

using the PyTorch[14] library, Hugging Face Transformers library, and Google Colab GPU. The Hugging Face library includes different classes for fine-tuning suited to a specific downstream task. For example, in this case, "BertForSequenceClassification" was used. This is the normal BERT model with an added single linear layer on top for classification. As the input data is fed, the entire pre-trained BERT model and the additional untrained classification layer are trained on this task.

4.4.2 Model Training

We randomly split our dataset into a training set (70%) and a test set (30%) and then performed Stratified 5-Fold Cross-Validation on the training set. This means that the training set is further divided into 5 subsets (or "folds"). We trained each model 5 times, each time using one of the different folds as the validation set and the remaining folds as the training set, while ensuring that each subset has the same proportion of class labels. The test set is not used during the cross-validation process and is kept aside for evaluating the final model's performance at the end.

We began by establishing a foundational baseline model, using Naive Bayes with Bag of Words (BoW) as the feature vectors. After establishing a baseline with Naive Bayes, we explored more advanced machine learning models using only the user-generated data - description and title features. This allowed us to establish a starting point for training and evaluate the contribution of additional features in the following experiments.

Then from the description feature, we derived additional features such as events, named entities, and keywords. By incorporating them, we wanted to capture more detailed information from the customer complaint texts, potentially improving the models' classification performance. By trying different feature combinations, we can measure how a particular feature affects the model's performance. This strategy was repeated for different text encoding strategies and classification algorithms.

4.4.3 Performance Evaluation

To measure how well the classifier predicts the class labels, the most commonly used evaluation metrics are accuracy, sensitivity (or recall), specificity, precision, and F1 [18].

Accuracy is a metric that measures the number of correctly classified data samples divided by the total number of data samples. It is defined as:

$$accuracy = (\frac{TP + TN}{TN + FP + TP + FN}) \tag{4.1}$$

[14] https://pytorch.org/

Where:

- TP (True Positive) is the number of correctly predicted positive instances.
- TN (True Negative) is the number of correctly predicted negative instances.
- FN (False Negative) is the number of instances wrongly predicted as negative when they are positive.
- FP (False Positive) is the number of instances wrongly predicted as positive when they are negative.

Sensitivity (or Recall) measures the proportion of actual positive instances that are correctly predicted as positive by the classifier. The formula for sensitivity is:

$$sensitivity = (\frac{TP}{TP + FN}) \tag{4.2}$$

Precision is the proportion of positive samples that are correctly identified (or true positive). It is defined as:

$$precision = (\frac{TP}{TP + FP}) \tag{4.3}$$

Specificity measures the proportion of actual negative instances that are correctly predicted as negative by the classifier.

$$specificity = (\frac{TN}{TN + FP)}) \tag{4.4}$$

The **F1 measure** is the harmonic mean of precision and recall. It provides a balance between the two measures and is useful when we want to consider both precision and recall simultaneously. It is defined as:

$$F1 = (\frac{2 * precision * recall}{precision + recall}) \tag{4.5}$$

A **confusion matrix** summarizes the performance of a classification model by showing the number of true positives, false positives, true negatives, and false negatives for each class in the dataset. The rows of the matrix represent the true labels of the data, while the columns represent the predicted labels. A confusion matrix is very helpful in identifying misclassified labels and types of errors. The diagonal elements of the matrix represent the number of correctly classified samples, while elements outside of the diagonal represent the number of incorrectly classified samples [18]. By reading a confusion matrix we can detect if the model is misclassifying certain types of classes more often than others.

4.4.4 Model Optimization

After adding new features and choosing the best model, we performed feature selection, hyperparameter tuning, and re-sampling techniques in order to improve the model's performance. In fact, when adding new features, it's possible that the model may overfit the training data, which means that it performs well on the training set but poorly on the validation and/or test sets. Moreover, having an imbalanced dataset can also affect the model's performance.

1. **Hyperparameter Tuning**

   When it comes to hyperparameter tuning for LinearSVC, we experimented with various combinations of hyperparameters, including regularization strength (C), loss function (loss), penalty type (penalty), dual formulation (dual), and maximum iteration limit (max_iter). The hyperparameter tuning technique used was Randomized Search Cross-Validation (available in the scikit-learn library). This is a more efficient approach when dealing with a large number of hyperparameters. The set of hyperparameters that achieved the highest F1 macro score was selected.

   For XGBoost, we explored different combinations of hyperparameters such as learning rate (eta), number of trees (n_estimators), maximum tree depth (max_depth), subsampling ratio (subsample), column subsampling ratio (colsample_bytree), minimum child weight (min_child_weight), gamma regularization term (gamma), L1 regularization term on the weights (reg_alpha), and L2 regularization term on the weights (reg_lambda). The hyperparameter tuning technique used was Bayesian optimization with the Optuna[15] library. XGBoost has a wide range of hyperparameters, so we need a more efficient approach. Bayesian optimization selects hyperparameters to test based on the results of previous trials, which helps in minimizing the number of experiments required to reach an optimal configuration. The set of hyperparameters that achieved the highest F1 macro score was selected.

   For BERTimbau, we tried different combinations of hyperparameters, such as maximum sequence length for tokenization, weight decay, learning rate, batch size, number of training epochs, and warmup steps. The hyperparameter tuning technique used was Bayesian optimization with the Optuna library. The set of hyperparameters that achieved the lowest validation loss was selected.

2. **Feature Selection**

   We applied L1 Regularization on SVM, which sets the less important feature's coefficients to zero. So, this works as a feature selection. We used Chi-Square Feature Selection for both SVM and XGBoost, by ranking features based on their Chi-Square statistics or p-values and keeping the top features with the highest significance. Although TF-IDF is not considered a traditional feature selection method, we can adjust parameters to control which words or n-grams are included in the TF-IDF representation based on their document

[15] https://optuna.org/

frequency within the corpus. This can select features that are more likely to be informative and relevant to our task.

Regarding the embedding representations, it does not make sense to apply traditional feature selection techniques because they are typically designed for datasets where each feature is a separate column, and in our approach, we take the mean of the embeddings for each text.

3. **Handle Class Imbalance**

   To handle class imbalance, we adjust the class weights in our machine learning algorithms to make them more sensitive to the minority classes. We also applied SMOTE using the imbalanced-learn library[16] in Python. We chose oversampling because our train dataset was already limited and undersampling could lose potential data.

4.5 Evaluation

After optimization, we selected the best model(s) and conducted a final evaluation on an independent test dataset. This evaluation helps assess the model's ability to generalize and make accurate predictions on new, unseen instances. Using the test dataset, we calculated the usual performance metrics. Additionally, we generated a confusion matrix to gain insights into the model's classification performance across different complaint classes. In addition to quantitative evaluation metrics, we performed error analysis and feature importance analysis to gain a deeper understanding of the model's behavior.

4.6 Deployment

The deployment stage is out of the scope of this research. However, the company can take the models, insights, and recommendations developed during the data mining process so far, and integrate them into the operational environment of the organization.

[16] https://imbalanced-learn.org/stable/

Chapter 5

Experiments and Results

This chapter presents the experiments and results of the application of the methodology described in the previous chapter, following the CRISP-DM framework. We conduct a series of experiments to address the research questions and objectives outlined in the earlier sections of this thesis:

1. What machine learning algorithms are most effective when applied to text classification problems?
2. How does the performance of traditional machine learning algorithms compare to deep learning models in classifying customer complaints?
3. What is the impact of using different feature encoding techniques, such as TF-IDF, word embeddings, or contextualized representations, on the accuracy of customer complaint classification?
4. Can the integration of keyword extraction using YAKE!, event extraction using text2story, and named entities improve the accuracy of customer complaint classification in Portuguese?
5. How can the findings from the exploratory analysis and text classification guide the development of more accurate labels for customer complaints, taking into account the identified patterns or themes in the data?

We use tables and figures to discuss and interpret the performance of each model, and any significant findings or trends observed.

5.1 Data Visualization

In this section, we focus on visually representing the processed data to gain further insights such as patterns or relationships within the complaints. By examining the patterns and themes of the texts, we can enhance our understanding of customer concerns and language.

To understand the importance of specific words within each class of our data, we used TF-IDF to encode the text, as it provides a quantitative measure of the importance of words in the entire dataset. To visualize the distribution of complaints in a lower-dimensional space and explore potential clusters between them, we used word embeddings.

5.1.1 Using TF-IDF encodings

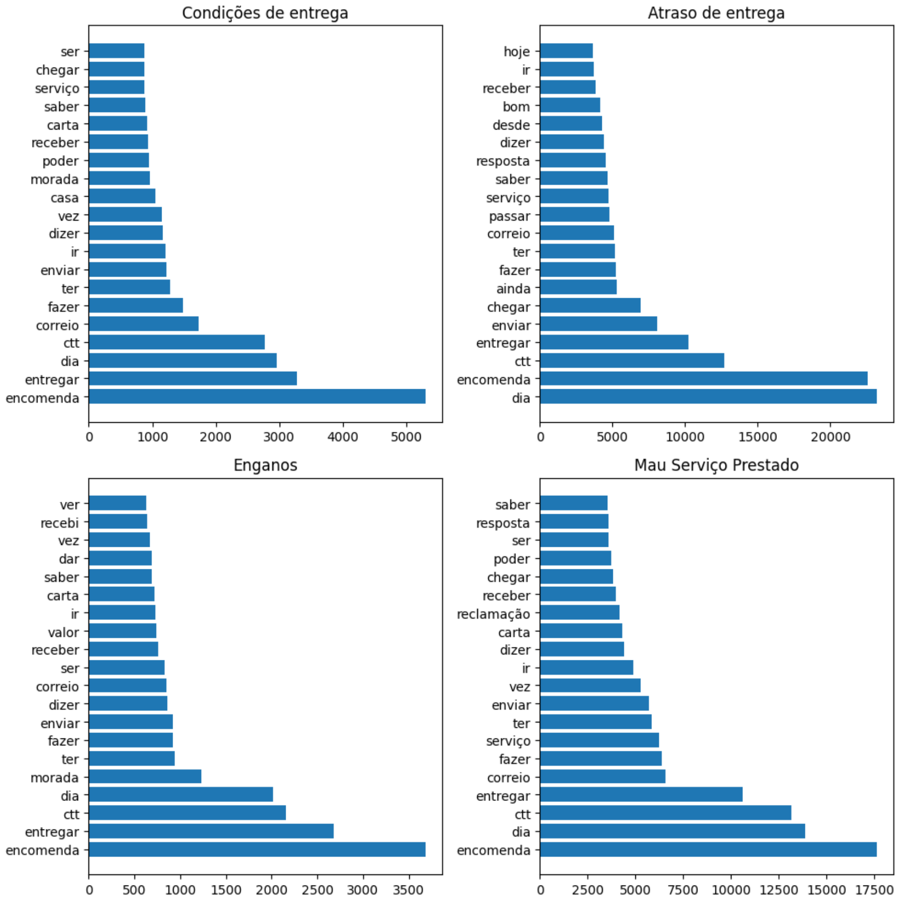


Figure 5.1: Most frequent words per Complaint Reason. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

Figure 5.1 displays the top 15 words per class based on the frequency of words in the texts belonging to each class, and Figure 5.2 displays the top 15 words per class based on TF-IDF scores. In both figures, we observe that some words appear in the top 15 lists of multiple classes, namely *"encomenda"*, *"entregar"*, *"ctt"* and *"correio"*, which makes sense based on the domain of our data. However, overall the words are very similar across classes, which could indicate an overlap or ambiguity between classes. The limited number of distinctive words might also indicate that word frequencies alone might not be sufficient to distinguish classes and that we need to introduce additional feature engineering and context of the words within the documents. Also, it is interesting to see that the word *"reclamação"* is more frequent in class *"Mau Serviço Prestado"*, which suggests that this class represents customers who are more likely to want to present a formal complaint.

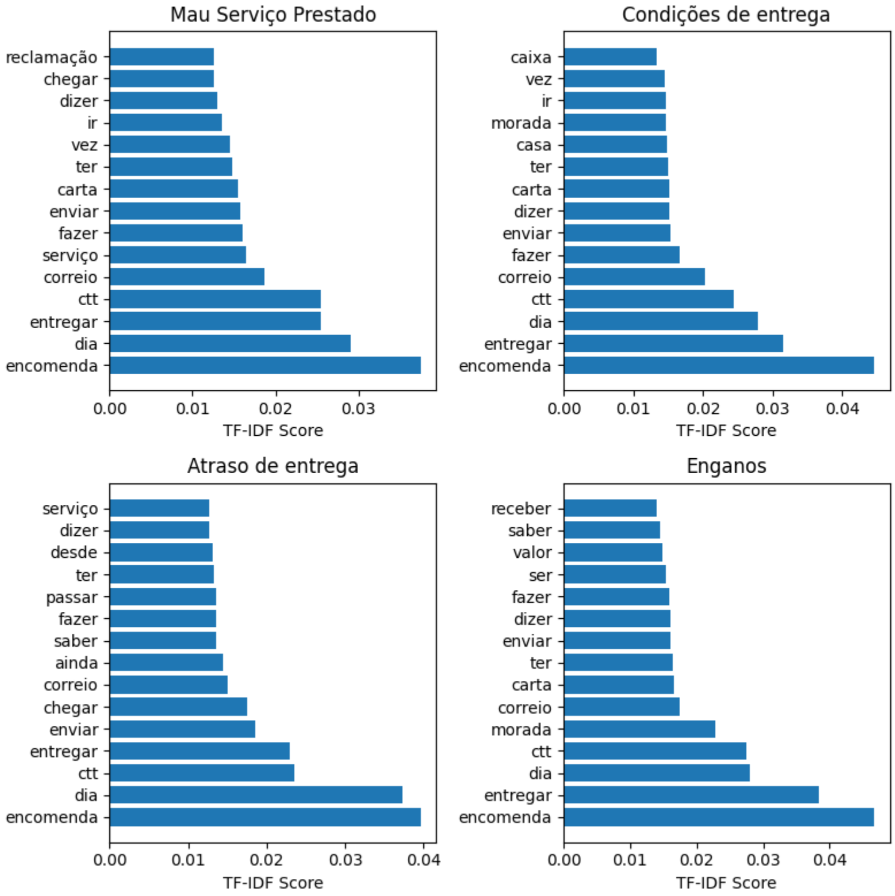


Figure 5.2: Top 15 words per class based on TF-IDF scores. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

Figure 5.3 shows the t-SNE visualization of the TF-IDF encoded texts for each class, with each class centroid. The centroids overlap, and there are no clearly visible clusters, which indicates that the texts are similar, due to shared vocabulary.

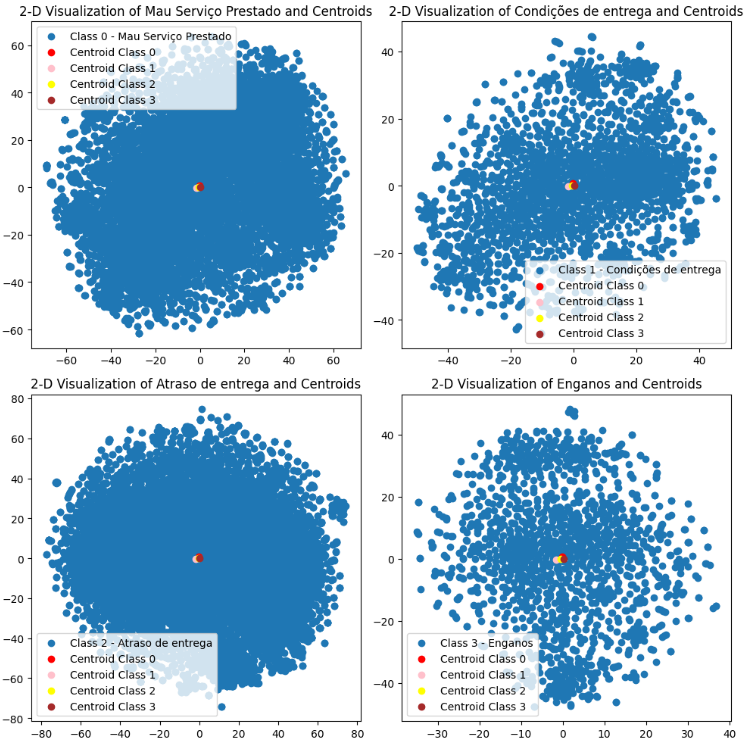


Figure 5.3: t-SNE Visualization of TF-IDF Encoded Texts with Class Centroids. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

In summary, these plots indicate the need for the use of more advanced NLP techniques like word embeddings or contextual models like BERT.

5.1.2 Using Embeddings

Figure 5.4 and Figure 5.5 display the t-SNE visualizations of the Glove and Word2Vec encoded texts for each class, respectively, along with their respective class centroids. The overlapping centroids suggest that the average representations of the texts within each class are relatively close to each other in the reduced t-SNE space, and the embeddings might not effectively separate the texts belonging to different classes based on the available features. In each plot, we see two distinct clusters within each class, which suggests that the texts within each class can have subgroups or variations in their semantic content.

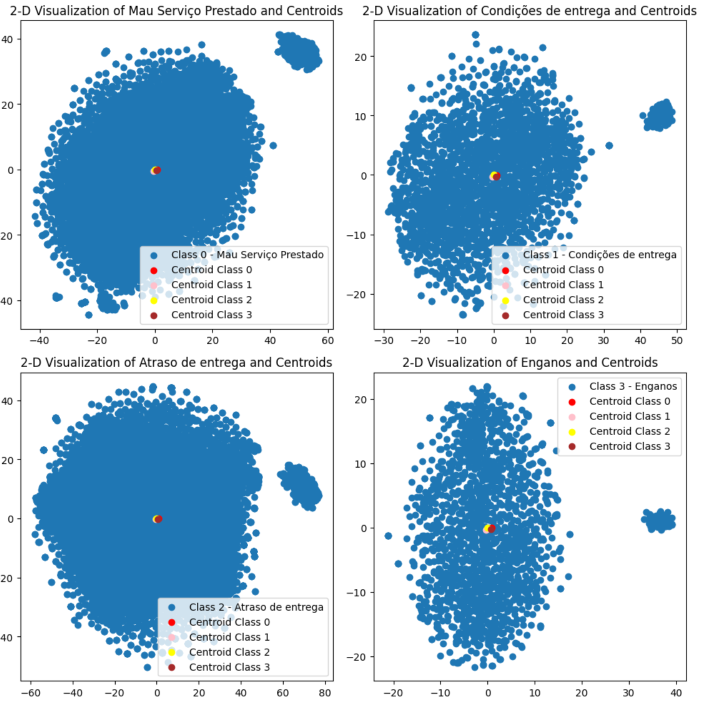


Figure 5.4: t-SNE Visualization of Glove Encoded Texts with Class Centroids. Labels: *“Condições de entrega”* - “Delivery Conditions”, *“Atraso de entrega* - “Delay in Delivery”, *“Enganos”* - “Mistakes”, *“Mau Serviço Prestado”* - “Poor Service Provided”.

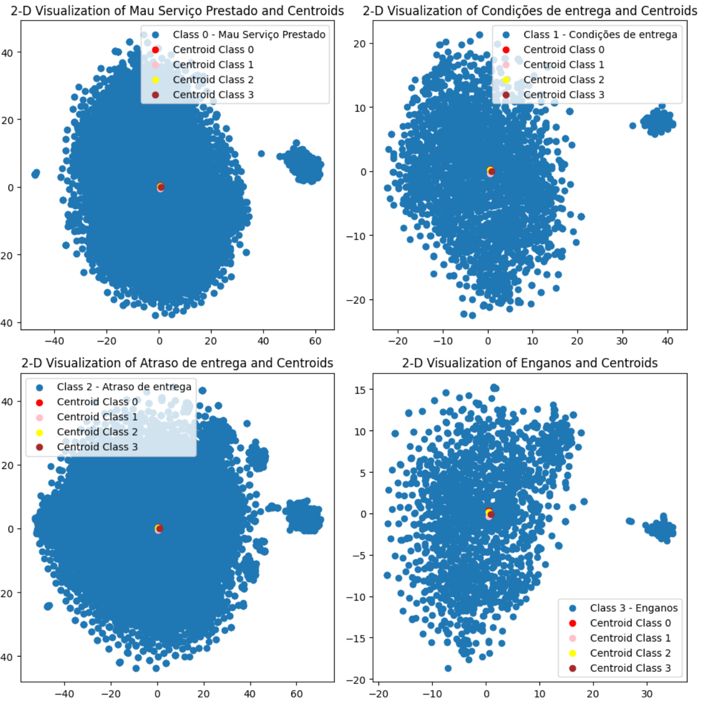


Figure 5.5: t-SNE Visualization of Word2Vec Encoded Texts with Class Centroids. Labels: *“Condições de entrega”* - “Delivery Conditions”, *“Atraso de entrega* - “Delay in Delivery”, *“Enganos”* - “Mistakes”, *“Mau Serviço Prestado”* - “Poor Service Provided”.

In summary, we foresee some challenges in the classification of these texts. The class ambiguity indicates that the decision boundaries between classes are not well-defined, making it challenging for a classifier to accurately distinguish between them. This ambiguity could lead to misclassifications, uncertainty, and lower classification accuracy.

5.2 Baseline Model

We started by building a baseline Multinomial Naive Bayes model using the original feature "description" and the Bag of Words (BoW) technique to encode the data. The purpose of establishing this baseline model is to measure the effectiveness of more complex models or advanced techniques in subsequent sections of the thesis. In our experiment, this model achieved an accuracy score of 0.63 and a macro F1-Score of 0.40 on the validation set. More detailed results of this experiment can be found in Table 7.1 of Appendix A.

5.3 Text Classification with TF-IDF using traditional ML methods

The objective of this section is to answer the Research Question 1, which aims to evaluate the performance of several machine learning algorithms when applied to our text classification problem. By examining their efficacy, we can identify the most effective machine learning algorithms for text classification and which algorithms are particularly well-suited for this task and can serve as a basis for selecting appropriate methods in future similar projects.

In this section we also investigate the impact of employing different feature encoding techniques, starting with TF-IDF, in order to answer Research Question 3.

5.3.1 Initial Experiments

We started by building initial models using the original features, namely description and/or title. These features contain user-provided information and are likely to include crucial details regarding the nature of the complaint. They play an important role in assessing the impact of additional features and model enhancements.

The size of n-grams in TF-IDF can affect the result of text classification. Larger n-gram sizes can capture more context, but can also lead to increased dimensionality and possibly overfitting. We conducted a grid search to find the best TF-IDF configuration for each model based on the validation macro F1-score. This involved fine-tuning the choice of n-gram range (unigrams, bigrams, trigrams), maximum document frequency to filter out terms that are too common, and minimum document frequency to eliminate rare and potentially noisy terms. More detailed results can be observed in Table 7.2 of Appendix A. For this task of classifying customer complaints using only the “description” feature, the best configuration uses both unigrams and bigrams, filters terms appearing in more than 80% of the documents, and considers a minimum document frequency of 1%.

To analyze the impact of adding “title” as a feature to the classification pipeline, we compared the results of the classifiers using both the “description” feature alone and the combination of “title” and “description”, using the same TF-IDF configurations as before.

Table 5.1 presents the validation accuracy and F1-Score for each classifier on two experiments: “description” and “description and title”. Random Forest and Gradient Boosting achieve the highest accuracy (0.64) when using the “description” feature, and also when using the “description and title” features (0.65). In terms of F1-Score, XGBoost and Linear SVC achieve the highest score (0.43) when using the “description” feature and the highest score (0.45) when using the “description and title” features. Random Forest exhibits high accuracy, but its F1-scores are relatively lower. This suggests that it may tend to predict the majority class more often, resulting in a higher accuracy but lower F1-score due to imbalanced class distribution. K-Nearest Neighbors, Decision Tree, and Naive Bayes achieve the worst F1-Scores. Moreover, we were able to increase the F1-Score from the Baseline Model, increasing it from 0.40 to 0.45.

Furthermore, the performance of the models improves with the addition of “title”, so it can be concluded that the “title” feature is informative and can help the classifiers discriminate between different classes. More detailed results can be found in Tables 7.3 and 7.4 of Appendix A.

Table 5.1: Initial Experiments with TF-IDF Vectorizer.

	Description		**Description and Title**	
Classifier	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
k-Nearest Neighbors	0.57	0.39	0.58	0.41
Decision Tree	0.50	0.35	0.51	0.36
Random Forest	**0.64**	0.36	**0.65**	0.38
Gradient Boosting	**0.64**	0.41	**0.65**	0.43
XGBoost	0.63	**0.43**	0.64	**0.45**
Naive Bayes	0.63	0.35	0.64	0.37
Linear SVC	0.63	**0.43**	0.64	**0.45**

In summary, the results suggest that the Linear SVC and XGBoost classifiers are the top-performing models among the tested classifiers, consistently achieving the highest F1-Score across both sets of features. This makes them the most promising candidates for further improvement in the next section.

5.3.2 Find Best Combination of Features

To determine whether the integration of keyword extraction using YAKE!, event extraction using text2story, and named entities improve the performance of customer complaint classification models (Research Question 4), we created a set of possible combinations of features from the features available (“description”, “title”, “keywords”, “events”, “entities”), and we used the same TF-IDF Vectorizer of the previous experiments.

For each combination, we trained and evaluated our classification models using the same metrics as in the previous experiments. The goal was to determine the most informative subset of features.

Based on the results summarized in Table 5.2 and Table 5.3, it seems that the integration of “keywords”, “events”, and “entities” did not provide significant additional value to the customer complaint classification task. Interestingly, in the case of LinearSVC, including “keywords” increased Precision by 1%, maintaining the same F1-Score and Recall.

Table 5.2: LinearSVC results using different feature combinations and TF-IDF Vectorizer.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.63	0.49	0.43	0.43
description, title	0.64	0.50	0.44	0.45
description, title, keywords	0.63	0.51	0.44	0.45
description, title, events	0.64	0.50	0.44	0.45
description, title, entities	0.64	0.50	0.44	0.45
description, title, keywords, events	0.63	0.50	0.44	0.44
description, title, keywords, entities	0.63	0.51	0.44	0.45
description, title, events, entities	0.64	0.50	0.44	0.45
description, title, keywords, events, entities	0.63	0.50	0.44	0.44

Table 5.3: XGBoost results using different feature combinations and TF-IDF Vectorizer.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.63	0.53	0.42	0.43
description, title	0.64	0.53	0.44	0.45
description, title, keywords	0.64	0.53	0.44	0.45
description, title, events	0.64	0.52	0.43	0.44
description, title, entities	0.64	0.52	0.43	0.44
description, title, keywords, events	0.64	0.51	0.43	0.44
description, title, keywords, entities	0.64	0.51	0.43	0.44
description, title, events, entities	0.65	0.53	0.44	0.44
description, title, keywords, events, entities	0.65	0.53	0.44	0.44

In the next section, we will explore the impact of feature selection, hyperparameter tuning, and resampling techniques. These techniques can help reduce overfitting, speed up training, improve model generalization, and more. Since XGBoost and Linear SVC achieve similar results, and since Linear SVC is simpler and more computationally efficient, we will be focusing on optimizing Linear SVC in the next section.

5.3.3 Optimization Strategies

At this stage, we performed feature selection, resampling, and hyperparameter tuning on the base model, which is LinearSVC with TF-IDF Vectorizer applied to "description and title" features. The results of the experiments in the validation set employing the LinearSVC are displayed in Table 5.4. The results suggest that hyperparameter tuning alone did not result in improvements, which could be related to the poor quality or quantity of the data for certain classes. In fact, addressing class imbalance through techniques like SMOTE and class weighting, combined with hyperparameter tuning, led to improvements in model performance, especially in terms of F1-score, which is a valuable metric for imbalanced datasets.

Before feature selection, the training data had 1,240 features. After applying chi-square feature selection to the base model, the number of features was reduced to 1,042, and after L1 regularization, which encourages feature selection, the number of features was reduced to 1,219. Regarding the techniques to deal with class imbalance, SMOTE generated synthetic samples for the minority classes, and LinearSVC automatically adjusted the class weights inversely proportional to the class frequencies in the training data.

Table 5.4: Optimization of LinearSVC with TF-IDF Vectorizer.

Strategy	**Accuracy**	**F1-Score**
Base Model	**0.64**	0.45
Base Model + Hyperparameter Tuning (HT)	0.63	0.45
SMOTE + Base Model	0.55	0.44
SMOTE + Base Model + HT	0.59	0.48
Class Weighting + Base Model	0.59	0.47
Class Weighting + Base Model + HT	0.63	**0.50**
L1 Regularization + Base Model	**0.64**	0.45
L1 Regularization + Base Model + HT	0.63	0.45
Chi-Square Feature Selection + Base Model	**0.64**	0.45
Chi-Square Feature Selection + Base Model + HT	0.63	0.45
Class Weighting + Base Model + HT + SMOTE	0.59	0.48
Class Weighting + Base Model + HT + L1 Regularization	0.63	**0.50**
Class Weighting + Base Model + HT + Chi-Square Feature Selection	0.63	**0.50**

Overall, the "Class Weighting + Base Model + HT" strategy yielded the best F1-Score results because it addressed the class imbalance problem while simultaneously optimizing the model's parameters. In summary, hyperparameter tuning is just one step in model optimization. To achieve the best results, it is often necessary to combine it with other techniques that address data preprocessing, feature engineering, and class imbalance.

5.3.4 Evaluation on Test Set

The best-performing model based on the highest F1-score on the validation set is LinearSVC with “Class Weighting + Base Model + HT”. We evaluated its performance on a test dataset and achieved an Accuracy of 0.63, Precision of 0.49, Recall of 0.52, and F1-Score of 0.50.

The diagonal elements (from top-left to bottom-right) in the confusion matrix (Figure 5.6) represent the true positives for each class, and they generally have higher values than off-diagonal elements. This suggests that the model is reasonably good at correctly classifying some of the samples, especially for class *“Atraso de Entrega”* (2,646 true positives) and class *“Mau serviço prestado”* (1,215 true positives). However, there is some class imbalance, with class *“Atraso de Entrega”* having a significantly higher number of samples compared to the other classes. Additionally, we can identify classes that are frequently confused with one another. For example, 686 instances of Class *“Mau serviço prestado”* were classified as *“Atraso de Entrega”* and 265 as Class *“Enganos”*. Class *“Condições de entrega”* is frequently misclassified across all classes, with Class *“Mau Serviço Prestado”* standing out. Class *“Atraso de Entrega”* has a relatively high number of misclassifications as class *“Mau serviço prestado”* (380 times). Finally, class *“Enganos”* is often also misclassified as class *“Mau serviço prestado”* (110 times).

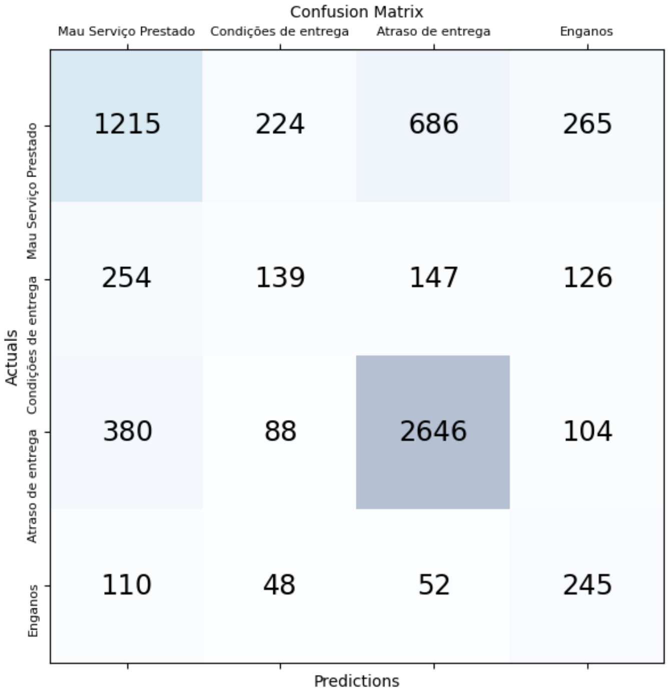


Figure 5.6: Confusion Matrix. Labels: *“Condições de entrega”* - “Delivery Conditions”, *“Atraso de entrega* - “Delay in Delivery”, *“Enganos”* - “Mistakes”, *“Mau Serviço Prestado”* - “Poor Service Provided”.

Figure 5.7 displays class-wise metrics for each complaint class. *"Atraso de entrega"* class has the highest Precision, indicating that when the model predicts this class, it is often correct. It also has a high Recall, suggesting that the model effectively identifies most of the instances of this class. *"Mau Serviço Prestado"* has a moderate Precision score, indicating that it correctly predicts this class reasonably well. However, its Recall score is relatively low, suggesting that it may miss many instances of this class. *"Enganos"* class has a relatively low Precision, indicating that when the model predicts this class, it has a higher chance of making mistakes. This trend is also evident in the confusion matrix, where the model correctly predicts 245 instances but makes at least 265 predictions that are incorrect, representing instances of false positives. However, its Recall score is higher compared to some other classes, indicating that it captures a good portion of the actual instances of this class. Finally, *"Condições de entrega"* class has the lowest performance in terms of these metrics.

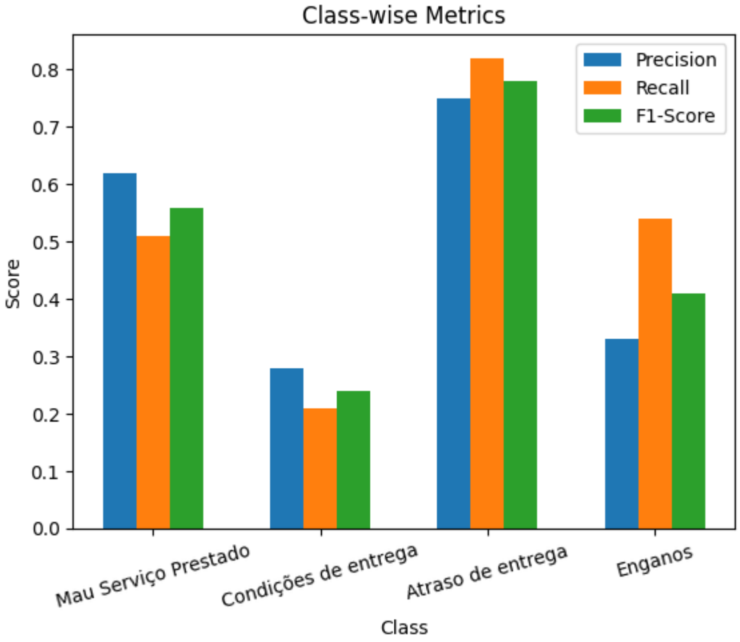


Figure 5.7: Metrics by Class. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

5.3.5 Model and Results Interpretation

At this stage, we explored different techniques to understand the model and its predictions. Figures 5.8 to 5.11 display the top and bottom features (words) that contribute the most and the least to classifying data points into each class based on the learned coefficients of our LinearSVC model. The length of the bars represents the magnitude of the coefficients, with longer bars indicating features that have a stronger influence on the model's decision for the class. Positive coefficients (green bars) contribute to the classification of the data points into the specified class, while negative coefficients (red bars) contribute to the classification into other classes.

Figure 5.8 reveals that the most important features to identify Class *"Condições de entrega"* are words like *"danificar"* (damage), *"condição"* (condition), *"levantar"* (pick up), *"espanto"* (amazement) and *"completamente"* (completely). Other words suggest that this class is associated with feedback related to delivery conditions, such as damaged packages (*"danificar"* (damage) and *"embalagem"* (packaging)), insufficient packaging (*"insuficiente"* (insufficient), or issues with delivery addresses (*"casa"* (house) and *"endereço"* (address)). On the other hand, words like *"atraso"* (delay), *"desde"* (since), *"dia"* (day), *"mês"* (month), *"ainda"* (yet), *"demor"* (take long), *"entregam"* (deliver), and *"serviço"* (service) are less relevant for this class, and seem to be related to Class *"Atraso de entrega"*.

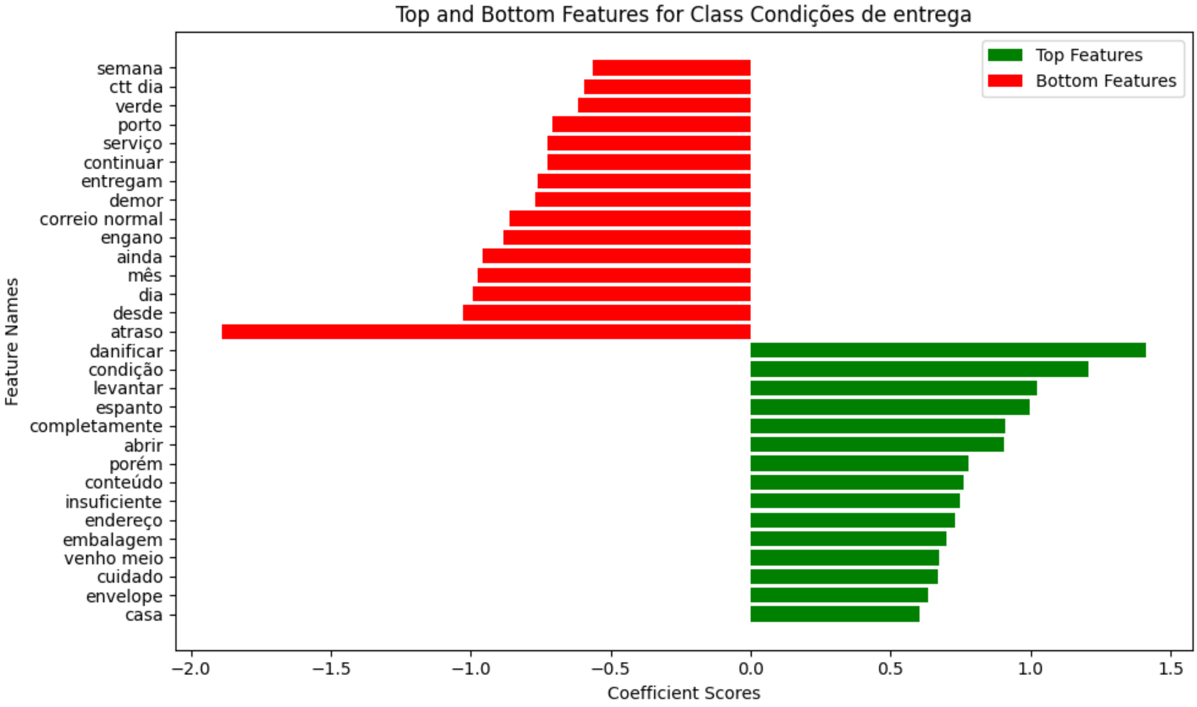


Figure 5.8: Top and Bottom Features for Class *"Condições de entrega"* (Delivery Conditions).

Figure 5.9 reveals that the most important features to identify Class *"Atraso de entrega"* are *"atraso"* (delay), *"dia"* (day), *"ainda"* (yet), *"desde"* (since) and *"demor"* (take long). Other words like "Portugal", *"atraso entregar"* (delay in delivery), *"ainda entregar"* (still not delivered), *"esperar"* (wait) and *"lisboa"* (Lisbon) indicate that concerns related to delivery delays in the Lisbon and Portugal areas are more common. On the other hand, words like *"morada"* (address), *"devolvir"* (return), *"casa"* (house), *"devolver"* (return) and *"danificar"* (damage) are less relevant for this class.

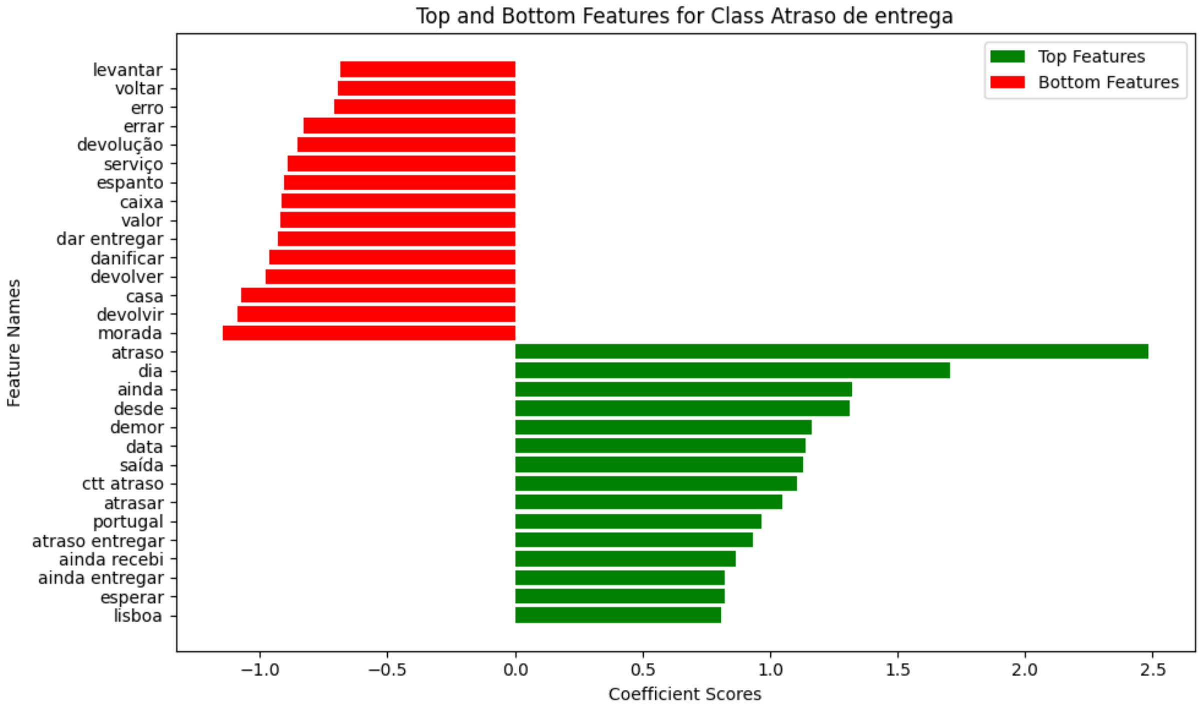


Figure 5.9: Top and Bottom Features for Class *"Atraso de entrega"* (Delay in Delivery).

Figure 5.10 reveals that the most important features to identify Class *"Enganos"* are words like *"engano"* (mistake), *"errar"* (mistake), *"enganar"* (mistake), *"cobrar"* (charge), *"erro"* (error), and *"valor"* (value). Together with words like *"encomenda entregar"* (package delivering) and *"outro pessoa"* (another person), they suggest issues related to mistakes, charges, and errors in the delivery process, to the wrong person. On the other hand, words like *"atraso"* (delay), *"demor"* (take long), *"data"* (date), *"danificar"* (damage) and *"ainda"* (yet) are less relevant for this class.

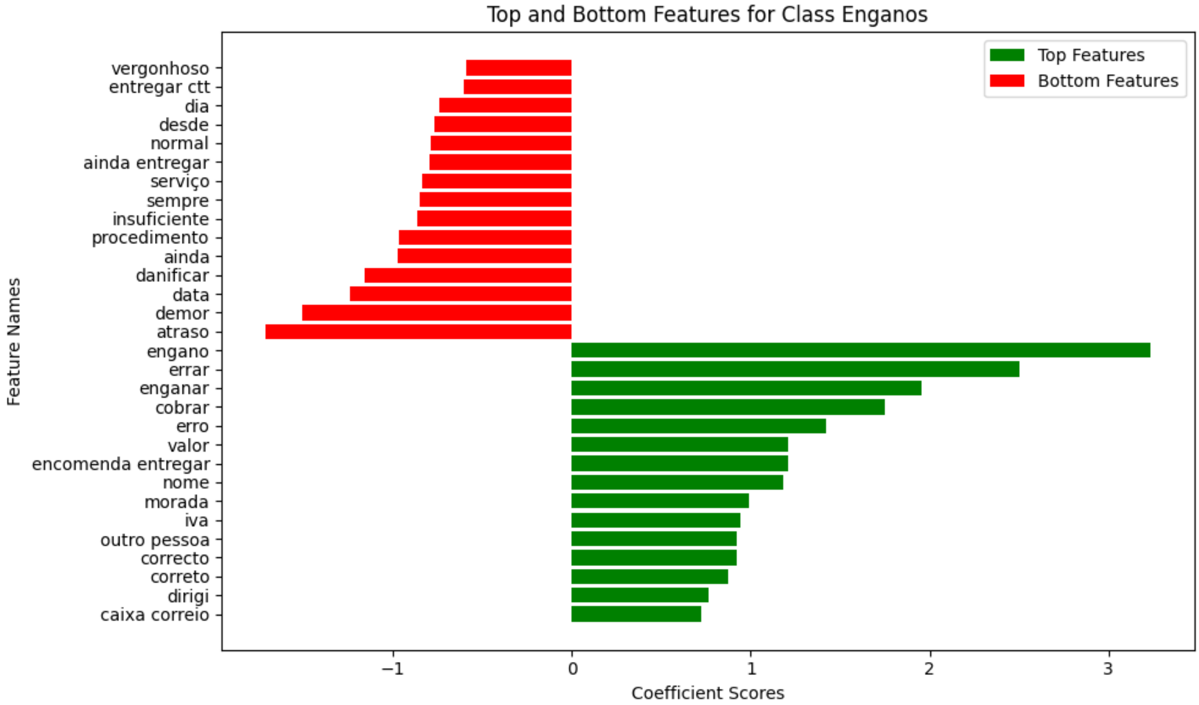


Figure 5.10: Top and Bottom Features for Class *"Enganos"* (Mistakes).

Finally, Figure 5.11 reveals that the most important features to identify Class *"Mau Serviço Prestado"* are words like *"devolvir"* (return), *"serviço"* (service), *"devolução"* (return), *"mau serviço"* (poor service) and *"profissionalismo"* (professionalism). Together with words like *"péssimo"* (terrible), *"funcionário"* (employee) and *"reembolso"* (refund), they indicate dissatisfaction with the service and requests for refunds or returns. On the other hand, words like *"atraso"* (delay), *"dia"* (day), *"ctt atraso"* (ctt delay), *"bom"* (good) and "encomenda" (package) are less relevant for this class.

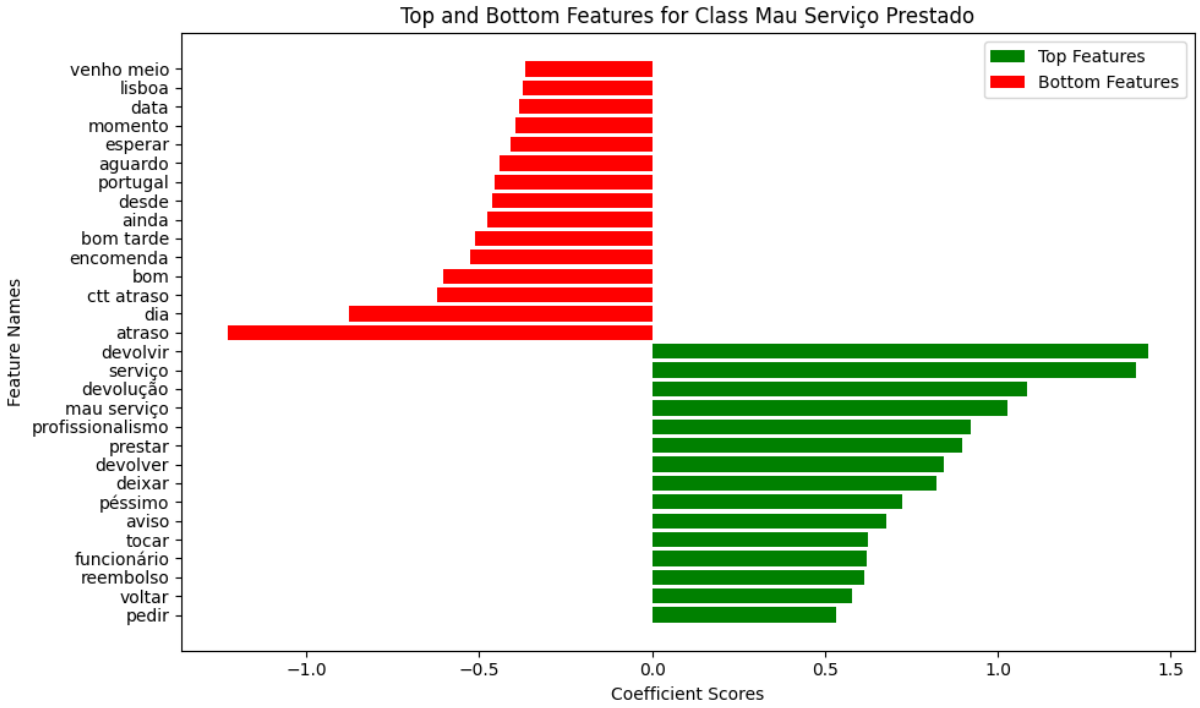


Figure 5.11: Top and Bottom Features for Class *"Mau Serviço Prestado"* (Poor Service Provided).

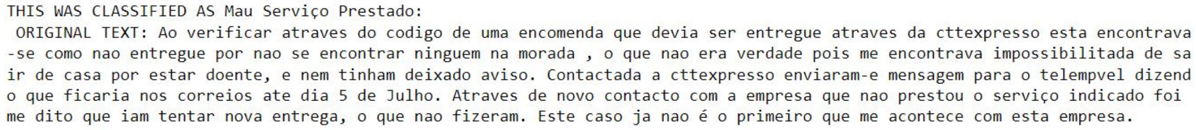
Some bottom features in each class, such as *"atraso"* (delay), *"demor"* (take long), *"ainda"* (yet), *"desde"* (since), and *"serviço"* (service) are common words that do not seem to provide strong discriminatory information for the specific class. This suggests that the presence of such words may not be helpful in distinguishing instances of that class. Interestingly, the presence of specific keywords in the top features suggests that the training data contains instances with strong language cues related to each class.

Following this analysis, we examined a sample of misclassified complaints to identify the most common errors and types of misclassifications. Looking at the confusion matrix in Figure 5.6, we see two main issues:

1. Classes "Poor Service Provided" and "Mistakes" seem to have lower true positives compared to the other two classes, and they are also more frequently misclassified into other classes. These two classes may need special attention, as they appear to be more challenging for our model.

2. Additionally, Class "Delay in Delivery" is often confused with class "Poor Service Provided", and vice-versa. This indicates that complaints related to delivery delays are often associated with complaints about poor service.

After conducting a manual analysis of a sample of misclassified examples, several patterns have emerged. Complaints labeled as "Mistakes" mostly relate to instances where letters or packages are sent to the incorrect recipient or address. Upon analyzing complaints originally categorized as “Mistakes”, but subsequently classified differently by the model, we observed that a lot of these complaints actually revolved around issues such as the mail carrier not ringing the doorbell and falsely claiming no one was home, or failing to provide a delivery notification when attempting to deliver a package or letter. The example in Figure 5.12 describes this situation. In this case, it appears that the true labels or categories assigned to the complaints might not accurately reflect the specific issues raised in the complaints themselves. Thus, it seems that the problem is with the true labels or the categorization of complaints rather than the model’s predictions.



```
THIS WAS CLASSIFIED AS Mau Serviço Prestado:
 ORIGINAL TEXT: Ao verificar atraves do codigo de uma encomenda que devia ser entregue atraves da cttexpresso esta encontrava
-se como nao entregue por nao se encontrar ninguem na morada , o que nao era verdade pois me encontrava impossibilitada de sa
ir de casa por estar doente, e nem tinham deixado aviso. Contactada a cttexpresso enviaram-e mensagem para o telempvel dizend
o que ficaria nos correios ate dia 5 de Julho. Atraves de novo contacto com a empresa que nao prestou o serviço indicado foi
me dito que iam tentar nova entrega, o que nao fizeram. Este caso ja nao é o primeiro que me acontece com esta empresa.
```

Figure 5.12: Example of a customer complaint with label “Mistakes”, but classified as “Poor Service Provided”. Translation: “When I checked using the tracking code of a package that was supposed to be delivered through CTT Expresso, it was marked as undelivered due to no one being present at the address, which was not true because I was unable to leave my home due to illness, and they had not even left a notice. After contacting CTT Expresso, they sent me a message to my mobile phone saying that the package would be held at the post office until July 5th. Upon further contact with the company that did not provide the promised service, I was told that they would attempt a new delivery, which they did not do. This is not the first time such an incident has occurred with this company.”.

Furthermore, it appears that the complaints that are labeled as “Poor Service Provided” and that were classified as “Delivery Delay” by the model actually encompass a wide range of problems related to bad service:

1. Lost Shipments: Some complaints involve the loss of packages during transit, where customers did not receive their orders at all.

2. Failure to Meet Expedited Delivery Promises: Customers claim to pay extra for expedited delivery services like *“correio azul”*, *“correio verde”* or *“correio registado”* (blue, green, or registered mail), expecting faster delivery. However, these services were not fulfilled.

3. Customs Issues: Complaints include problems with customs, such as parcels being held at customs without notification to the recipient, or packages not progressing for delivery even after customs fees were paid.

4. Incorrect Status: Some customers reported that their orders were marked as “delivered” in the tracking system, but they had not actually received their items.

5. Lack of Street Service: Lack of adequate mail service on certain streets has caused important documents like bills, pensions, fines, and crucial information to arrive late.

6. Unhelpful Customer Support: Customers have expressed dissatisfaction with the customer support, stating that it did not adequately address their issues or provide solutions and that it consisted of automated responses.

7. Payment Collection Problems: Complaints also include situations where orders were sent with a *"envio à cobrança"* ("cash on delivery") option, but the payment made by the recipient did not reach the sender.

It also appears that the complaints that are labeled as "Delivery Delay" and that were classified by the model as "Poor Service Provided" also cover the same issues listed above. The classifier likely mistakes them because both of them contain similar keywords or phrases, making it challenging for the classifier to distinguish between them. For example, phrases like "since day", "delay", "package", "still have not received", "never arrived", and "ctt" are present in both types of complaints. Furthermore, certain complaints can fall into both of these categories simultaneously, and this can be a source of confusion for text classification models.

5.3.6 Summary

Table 5.5: Results summary.

Model	**Features**	**Accuracy**	**F1-Score**
BoW + Naive Bayes (Baseline Model)	description	0.63	0.40
TF-IDF + Linear SVC	description	0.63	0.43
TF-IDF + XGBoost	description	0.63	0.43
TF-IDF + Linear SVC	description, title	**0.64**	0.45
TF-IDF + XGBoost	description, title	**0.64**	0.45
TF-IDF + Linear SVC + Class Weighting + HT	description, title	0.63	**0.50**

So far, we can conclude that at least when using TF-IDF to encode text data, adding new features such as keywords, named entities, and events, does not improve the performance of our classification models. Considering that the new features were extracted from the "description" feature, it is possible that the additional features introduced redundant information, and that the "description" and "title" features contain the most informative information for the classification task. Another possibility is the missing values in the new features (especially entities and events) that we filled with empty strings (""). This is one approach to handle missing values, but it may not always be the most appropriate solution. Finally, there is a chance that the keyword extraction method, event extraction technique, or NER algorithm failed to capture all relevant terms.

We were able to improve the classification performance from an F1-Score of 0.40 to 0.50. The best classification model for TF-IDF features is LinearSVC or XGBoost. They both achieved similar results on the validation set. This could be explained by the fact that they both handle well high-dimensional data, and they both offer mechanisms (L1 and L2 regularization) to control overfitting. However, Linear SVC is faster, making it a better choice for large text datasets where training time is a concern.

Moreover, we were able to identify common sources of complaints, ranging from lost shipments and failed promises of expedited delivery to customs problems, incorrect delivery status updates, lack of street service, unhelpful customer support, and issues with payment collection. We can conclude that complaints from classes "Poor Service Provided" and "Delivery Delay" often overlap and that the true labels or categories assigned to the complaints might not accurately reflect the specific issues raised in the complaints themselves, more specifically in class "Mistakes".

5.4 Text Classification with Word Embeddings using traditional ML methods

In the previous section, we presented the results obtained using TF-IDF as a text encoding technique. Building upon those findings, we now explore the efficacy of word embeddings in improving the F1-Score of classification. The goal is to compare the performance of different text encoding techniques when classifying customer complaints (Research Question 3). Unlike TF-IDF, which represents words based on their frequency and inverse document frequency, word embeddings capture the semantic and contextual relationships between words.

5.4.1 Initial Experiments

Similarly to the Text Classification with TF-IDF, we started by building initial models using the original features, namely description and/or title.

The table 5.6 presents the performance of various classifiers using Word2Vec (Skip-Gram 600) pre-trained embeddings. More detailed results of these experiments are shown in Tables 7.5 and 7.6 in Appendix A. When considering only the "description" feature, the highest accuracy achieved is 0.64, which is obtained by Linear SVC. The highest F1-Score achieved is 0.40, which is obtained by Gradient Boosting. When considering both the "description" and "title" features, the highest accuracy achieved is 0.65, which is obtained by Linear SVC. The highest F1-Score achieved is 0.42, which is obtained by both Gradient Boosting and XGBoost. In this scenario, Decision Tree, Naive Bayes, and KNN show relatively lower performance compared to other classifiers. Gradient Boosting and XGBoost are the best models, with competitive accuracy and F1-Score values, which makes them the most promising candidates for further improvement in the next section.

Table 5.6: Initial Experiments using Word2Vec Embeddings.

	Description		**Description and Title**	
Classifier	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
k-Nearest Neighbors	0.50	0.32	0.53	0.34
Decision Tree	0.46	0.32	0.47	0.33
Random Forest	0.61	0.33	0.62	0.34
Gradient Boosting	0.63	**0.40**	0.64	**0.42**
XGBoost	0.63	0.39	0.64	**0.42**
Naive Bayes	0.59	0.31	0.59	0.31
Linear SVC	**0.64**	0.38	**0.65**	0.41

The table 5.7 presents the performance of various classifiers using GloVe (600 dimensions) pre-trained embeddings. More detailed results of these experiments are shown in Tables 7.7 and 7.8 in Appendix A. When considering only the "description" feature, the highest accuracy achieved is 0.64, which is obtained by Linear SVC. The highest F1-Score achieved is 0.40, which is also obtained by Linear SVC. When considering both the "description" and "title" features, the highest accuracy achieved is 0.65, which is obtained by Linear SVC. The highest F1-Score achieved is 0.42, which is also obtained by Linear SVC. Linear SVC consistently performs very well, achieving the highest accuracy and F1-Score among all classifiers in both scenarios. Gradient Boosting and XGBoost also perform well. Similarly to the previous table, Decision Tree, KNN, and Naive Bayes show relatively lower performance compared to other classifiers.

Table 5.7: Initial Experiments using GloVe Embeddings.

	Description		**Description and Title**	
Classifier	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
k-Nearest Neighbors	0.50	0.32	0.52	0.33
Decision Tree	0.45	0.32	0.46	0.32
Random Forest	0.61	0.33	0.61	0.34
Gradient Boosting	0.63	0.39	0.64	0.40
XGBoost	0.63	0.39	0.64	0.40
Naive Bayes	0.57	0.30	0.58	0.31
Linear SVC	**0.64**	**0.40**	**0.65**	**0.42**

In summary, when using embeddings, Linear SVC, Gradient Boosting, and XGBoost are the top-performing classifiers. Both Word2Vec and GloVe embeddings show relatively similar trends in terms of classifier performance. Adding the "title" feature consistently improves the classifiers' performance across both tables, even if it is by 1%.

5.4.2 Find Best Combination of Features

The initial models use only the original features, namely description and title, that were generated by the customer. In this section, we explore the inclusion of additional features on the best models of the previous experiment.

Table 5.8 and Table 5.9 show the results of XGBoost and Gradient Boosting, respectively, when using Word2Vec pre-trained embeddings. In both cases, adding additional features beyond “description” and “title” does not improve the model’s performance.

Table 5.8: XGBoost results using different feature combinations and Word2Vec embeddings.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.63	0.50	0.40	0.40
description, title	0.64	0.53	0.42	0.42
description, title, keywords	0.63	0.52	0.41	0.41
description, title, events	0.63	0.51	0.41	0.41
description, title, entities	0.64	0.51	0.41	0.41
description, title, keywords, events	0.63	0.52	0.40	0.40
description, title, keywords, entities	0.63	0.53	0.41	0.41
description, title, events, entities	0.63	0.53	0.41	0.41
description, title, keywords, events, entities	0.63	0.51	0.40	0.40

Table 5.9: Gradient Boosting results using different feature combinations and Word2Vec embeddings.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.63	0.49	0.40	0.40
description, title	0.64	0.51	0.42	0.42
description, title, keywords	0.63	0.50	0.41	0.40
description, title, events	0.63	0.50	0.41	0.41
description, title, entities	0.64	0.51	0.42	0.42
description, title, keywords, events	0.63	0.50	0.41	0.41
description, title, keywords, entities	0.63	0.50	0.40	0.40
description, title, events, entities	0.63	0.51	0.41	0.41
description, title, keywords, events, entities	0.63	0.48	0.40	0.40

Table 5.10 and Table 5.11 show the results of LinearSVC and XGBoost, respectively, when using GloVe pre-trained embeddings. Once again, additional features beyond “description” and “title” do not improve the performance of our models.

Table 5.10: LinearSVC results using different feature combinations and Glove embeddings.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.64	0.52	0.41	0.40
description, title	0.65	0.55	0.42	0.42
description, title, keywords	0.64	0.53	0.42	0.42
description, title, events	0.65	0.55	0.42	0.42
description, title, entities	0.65	0.54	0.42	0.42
description, title, keywords, events	0.64	0.53	0.42	0.41
description, title, keywords, entities	0.64	0.53	0.42	0.41
description, title, events, entities	0.65	0.54	0.42	0.42
description, title, keywords, events, entities	0.64	0.52	0.42	0.41

Table 5.11: XGBoost results using different feature combinations and Glove embeddings.

Features	**Accuracy**	**Precision**	**Recall**	**F1-Score**
description	0.63	0.51	0.40	0.39
description, title	0.64	0.52	0.41	0.40
description, title, keywords	0.63	0.52	0.40	0.40
description, title, events	0.63	0.52	0.41	0.40
description, title, entities	0.63	0.51	0.41	0.40
description, title, keywords, events	0.63	0.53	0.40	0.40
description, title, keywords, entities	0.63	0.52	0.40	0.40
description, title, events, entities	0.63	0.52	0.41	0.41
description, title, keywords, events, entities	0.63	0.53	0.40	0.40

In summary, the addition of additional features like "keywords", "events", and "entities" does not lead to significant improvements in classification performance, so we will not be considering them in the next sections. This could be due to the relevance and quality of these additional features or potential issues related to feature engineering. In the next section, we will explore the impact of feature selection, hyperparameter tuning, and resampling techniques. For Word2Vec, we will be focusing on optimizing XGBoost. Both XGBoost and Gradient Boosting achieved the same results, but XGBoost is more computationally efficient and it has regularization parameters. For GloVe, we will be optimizing LinearSVC, which achieved higher F1-Scores compared to XGBoost.

5.4.3 Optimization Strategies

The quality of word embeddings can be sensitive to hyperparameter settings during training. At this stage, we performed hyperparameter tuning and considered techniques to handle class imbalance, like oversampling and class weighting.

In the case of XGBoost with Word2Vec embeddings (Table 5.12) hyperparameter tuning alone improved the accuracy to 0.65, while the F1-score increased to 0.44. Incorporating SMOTE with the base model (XGBoost trained with “description” and “title” features) led to a decrease in accuracy but an improvement in the F1 score. Combining SMOTE with hyperparameter tuning resulted in further improvements, with an accuracy of 0.62 and an F1-score of 0.48. The class weighting strategy showed improvements when combined with hyperparameter tuning, yielding an F1-score of 0.49.

Table 5.12: XGBoost with Word2Vec Embeddings Optimization.

Strategy	**Accuracy**	**F1-Score**
Base Model	0.64	0.42
Base Model + Hyperparameter Tuning (HT)	**0.65**	0.44
SMOTE + Base Model	0.61	0.45
SMOTE + Base Model + HT	0.62	0.48
Class Weighting + Base Model	0.63	0.44
Class Weighting + Base Model + HT	0.62	**0.49**
Class Weighting + Base Model + HT + SMOTE	0.63	0.48

Regarding LinearSVC with GloVe embeddings (5.13), hyperparameter tuning maintained the accuracy at 0.65 but improved the F1-score to 0.45. The application of SMOTE to the base model (LinearSVC with “description” and “title” features) led to a decrease in accuracy but a slight increase in the F1-score. Combining SMOTE with hyperparameter tuning resulted in an F1-score of 0.47. Introducing class weighting enhanced the F1-score to 0.49. The use of L1 regularization maintained an accuracy of 0.65 but showed a similar F1-score improvement as hyperparameter tuning alone.

Table 5.13: LinearSVC with GloVe Embeddings Optimization.

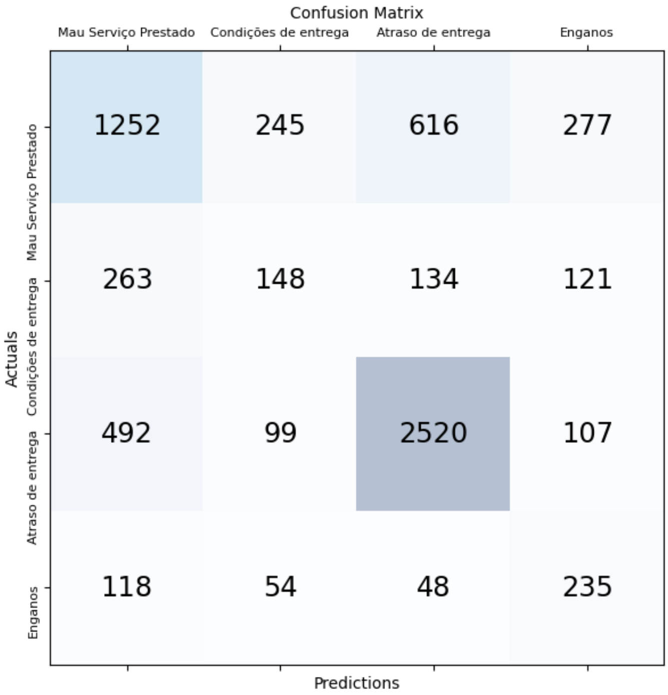
Strategy	**Accuracy**	**F1-Score**
Base Model	**0.65**	0.42
Base Model + Hyperparameter Tuning (HT)	**0.65**	0.45
SMOTE + Base Model	0.57	0.46
SMOTE + Base Model + HT	0.57	0.47
Class Weighting + Base Model	0.62	**0.49**
Class Weighting + Base Model + HT	0.62	**0.49**
L1 Regularization + Base Model	**0.65**	0.42
L1 Regularization + Base Model + HT	0.64	0.45
Class Weighting + Base Model + SMOTE	0.57	0.46
Class Weighting + Base Model + L1 Regularization	0.62	**0.49**

We conclude that, for both the XGBoost with Word2Vec embeddings and LinearSVC with GloVe embeddings models, the combination of class weighting with hyperparameter tuning (HT) achieved the best F1-scores among the strategies tested. Hyperparameter tuning optimizes the model's internal parameters to improve its overall performance, while class weighting addresses class imbalance, particularly benefiting the recall of minority classes.

5.4.4 Evaluation on Test Set

Both methods achieve similar results, but since LinearSVC is more interpretable than ensemble methods like XGBoost, we decided to choose it to evaluate on the test set. We achieved an Accuracy of 0.62, Precision of 0.48, Recall of 0.51, and F1-Score of 0.49 on the test set.

With the help of the confusion matrix in Figure 5.13, we can identify patterns of misclassification by the model. 616 instances of Class *"Mau serviço prestado"* were classified as *"Atraso de Entrega"*, and 492 instances of class*"Atraso de Entrega"* were predicted as *"Mau serviço prestado"*. This indicates that these two classes are often interchanged. Moreover, 263 instances of Class *"Condições de entrega"* were classified as *"Mau serviço prestado"*, which suggests that complaints about delivery conditions may be perceived as issues with service quality. Finally, 118 instances of Class *"Enganos"* were classified as *"Mau serviço prestado"*, which could mean that customers who experience mistakes in their orders might interpret them as a form of poor service.



	Mau Serviço Prestado	Condições de entrega	Atraso de entrega	Enganos
Mau Serviço Prestado	1252	245	616	277
Condições de entrega	263	148	134	121
Atraso de entrega	492	99	2520	107
Enganos	118	54	48	235

Figure 5.13: Confusion Matrix. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

Figure 5.14 displays class-wise metrics for each complaint class. *"Atraso de entrega"* class has the highest Precision, indicating that when the model predicts this class, it is often correct. It also has a high Recall, suggesting that the model effectively identifies most of the instances of this class. The model's performance for the *"Mau Serviço Prestado"* class is relatively balanced, with moderate precision, recall, and F1 score. *"Enganos"* class has a relatively low Precision, but higher Recall and F1-Score, indicating that while the model may identify some of these complaints, it also has a significant number of false positives. Finally, *"Condições de entrega"* class has the lowest performance in terms of these metrics.

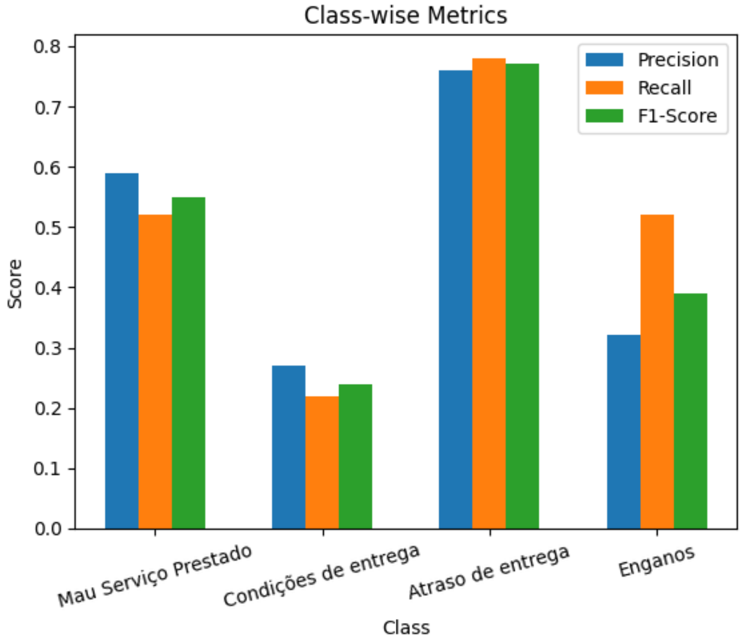


Figure 5.14: Metrics by Class. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

5.4.5 Model and Results Interpretation

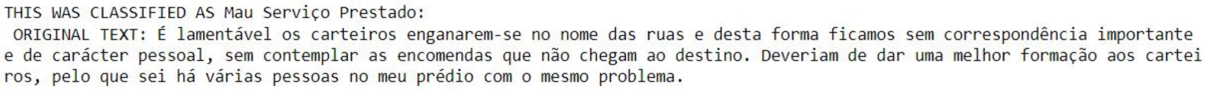
When working with averaged word embeddings, the interpretation becomes more abstract since each dimension in the embedding space does not directly correspond to a specific word. Therefore, in order to understand the model's predictions and the main causes of errors, we conducted a manual analysis of a sample of misclassified texts. Based on the confusion matrix (Figure 5.13), Class *"Mau Serviço Prestado"* ("Poor Service Provided") and *"Atraso de entrega"* ("Delivery Delay") seem to be confused together the most. Classes *"Condições de entrega"* ("delivery conditions") and *"Enganos"* ("Mistakes") still exhibit lower true positives and higher confusion with other classes.

Therefore, we analyzed specific misclassified examples within these classes. It appears that the model often confuses class "Poor Service Provided" and "Delivery Delay" due to the same reasons mentioned in section 5.3.5 - from delivering to the wrong destination, lack of service, customs issues, incorrect status, and others. However, now the complaints tend to be longer and more detailed. The customer tends to write a long story with dates and past interactions with the postal service, they mention what led to the “delivery delay” (issues from section 5.3.5), and they often express dissatisfaction with the response from the postal service.

In relation to complaints categorized as “Delivery Conditions” and accurately identified as such, they concern instances where:

- the delivery person failed to leave delivery notices
- delivery person did not ring the doorbell or attempt to deliver the package but instead left a notice claiming that the customer did not answer the door
- packages were delivered to the wrong person or address
- packages or letters arrived damaged or open
- mail was left outside of mailboxes

However, the “Delivery Conditions” misclassified complaints are often confused with “Poor Service Provided” and refer to similar issues to those stated above. Moreover, complaints labeled as “Mistakes” refer to packages or letters being sent to the wrong address or person. “Mistakes” misclassified complaints are also often confused with “Poor Service Provided”. For example, the text in figure 5.12 labeled as “Mistakes” was also classified as “Poor Service Provided” by this new classifier. Another example is given in Figure 5.15. This complaint was labeled as “Mistakes” but classified as “Poor Service Provided”. This complaint could potentially be labeled as both “Mistakes” and “Poor Service Provided” because it describes mistakes made by postal workers (errors in street names) that contribute to poor service quality. “Poor Service Provided” seems to be a broader category that encompasses various aspects of service quality, including mistakes made by postal workers.



```
THIS WAS CLASSIFIED AS Mau Serviço Prestado:
 ORIGINAL TEXT: É lamentável os carteiros enganarem-se no nome das ruas e desta forma ficamos sem correspondência importante
e de carácter pessoal, sem contemplar as encomendas que não chegam ao destino. Deveriam de dar uma melhor formação aos cartei
ros, pelo que sei há várias pessoas no meu prédio com o mesmo problema.
```

Figure 5.15: Example of a customer complaint with label “Mistakes”, but classified as “Poor Service Provided”. Translation: “It’s regrettable that the postal workers make mistakes in the street names, resulting in us missing out on important and personal correspondence, not to mention the packages that don’t reach their destination. They should provide better training to the postal workers; as far as I know, there are several people in my building facing the same issue.”.

5.4.6 Summary

Table 5.14: Results summary.

Model	Features	Accuracy	F1-Score
BoW + Naive Bayes (Baseline Model)	description	0.63	0.40
TF-IDF + Linear SVC	description	0.63	0.43
TF-IDF + Linear SVC	description, title	0.64	0.45
TF-IDF + Linear SVC + Class Weighting + HT	description, title	0.63	**0.50**
Word2Vec + XGBoost	description	0.63	0.40
Word2Vec + XGBoost	description, title	0.64	0.42
Word2Vec + Class Weighting + XGBoost + HT	description, title	0.62	0.49
GloVe + Linear SVC	description	0.64	0.40
GloVe + Linear SVC	description, title	**0.65**	0.42
GloVe + Class Weighting + Linear SVC	description, title	0.62	0.49

Once again, we can conclude that, when using averaged word embeddings to encode text data, adding new features such as keywords, named entities, and events, does not improve the performance of our classification models. However, adding the title feature slightly helps. The same happened when using TF-IDF to encode the texts.

When using averaged word embeddings to encode texts, LinearSVC and XGBoost achieved the highest accuracy and F1-scores on the validation set, among the tested classifiers. The same happened when using TF-IDF to encode the texts.

Moreover, we were able to identify common sources of complaints of type “Delivery Conditions”, ranging from packages being delivered to the wrong address, to broken packages, and instances where the delivery person failed to leave a delivery notice, and others. We conclude that complaints from classes “Poor Service Provided” and “Delivery Conditions” often overlap and that the complaints might raise more than one issue. In conclusion, the overall challenges and confusion patterns in this model appear to be consistent with the previous model using TF-IDF features, indicating that the underlying issues might be related to the nature of the data and labeling method or the complexity of the task rather than specific to the model architecture.

5.5 Language Models

In the previous section, we presented the results obtained using traditional machine learning algorithms for classifying customer complaints. In this section, we extend our analysis by evaluating the performance of deep learning models, namely BERT, for the same task. The purpose is to assess whether deep learning models exhibit superior performance compared to traditional machine learning algorithms (Research Question 2). By comparing the results of both approaches, we can gain insights into the potential benefits of employing deep learning techniques in the domain of customer complaint classification.

5.5.1 Initial Experiments

In this section, we present the results obtained from our experiments with BERT for text classification. We started by doing hyperparameter tuning of BERT using only "description" as a feature. A learning rate of approximately 8.24e-06, batch size of 32, maximum sequence length of 128, weight decay of approximately 0.076, AdamW optimizer, linear learning rate schedule with a warm-up phase of 20% of the total training steps, gradient clipping with a threshold of 1.0 and training for 15 epochs achieved the lowest loss of 0.86. Figure 5.16 displays an optimization history plot. The y-axis represents the value of the objective function - minimize the validation loss, and the x-axis represents the iterations. While we have minimized the validation loss, we see scattered loss values without clear convergence. Additionally, in Figure 5.17 we see that learning rate has the most significant effect on the model's performance.

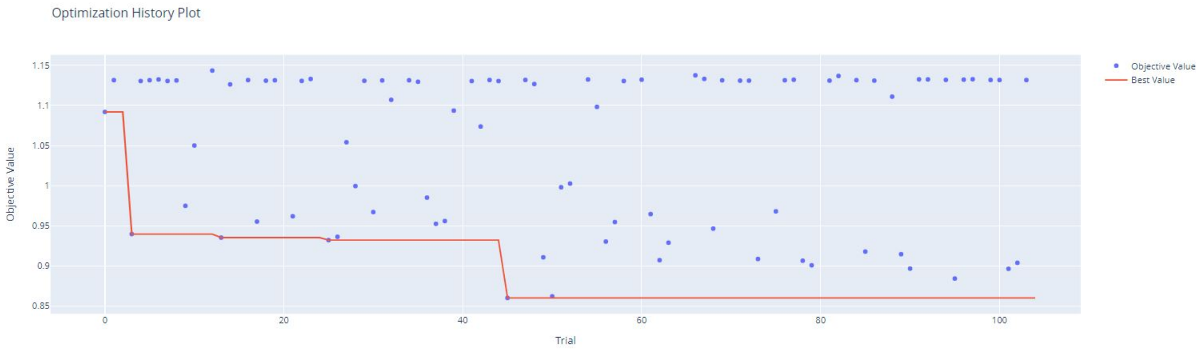


Figure 5.16: Optimization history plot.

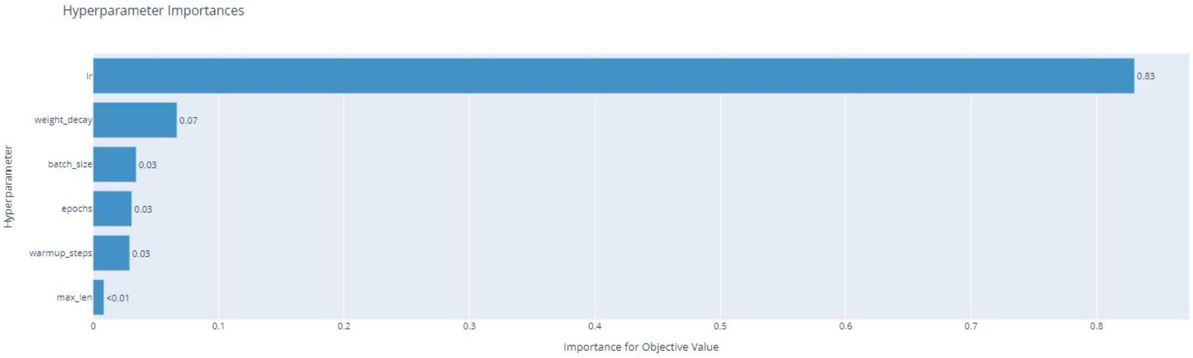


Figure 5.17: Parameter importances.

Then, we added the feature "title" as well, and we increased the maximum sentence length to 256 to accommodate the titles. As we see in Table 5.15, when considering only the "description" feature, the model achieved an Accuracy of 0.62 and F1-Score of 0.45. When considering both "description" and "title" features, the Accuracy increased to 0.66, and the F1-Score increased to 0.49. Therefore, adding the "title" feature improved the classification performance of our model.

Table 5.15: BERTimbau Validation Set Results.

Features	Accuracy	Precision	Recall	F1-Score
description	0.62	0.50	0.47	0.45
description, title	0.66	0.55	0.50	0.49

5.5.2 Find Best Combination Of Features

In this section, we wanted to explore the benefits of using additional features, such as “keywords”, “events”, and “entities”. To accommodate these extra features, we adjusted the maximum sentence length to 256 when adding one feature and 300 when adding multiple features while maintaining the previously tuned hyperparameters. During the experimentation phase, we encountered memory limitations when attempting to increase the maximum sentence length beyond 300 tokens due to out-of-memory errors. Nevertheless, we believe that further increasing the maximum sentence length could potentially enhance the model’s performance, especially when considering a combination of all features.

Table 5.16 summarizes the results of experiments with different feature combinations using BERTimbau. It appears that adding more features, such as “title”, “keywords”, “events”, and “entities”, tends to enhance the model’s performance across all metrics. The combination of description, title, and events seems to provide the best trade-off between precision, recall, and F1-score, with an accuracy of approximately 0.67 and an F1-Score of 0.50. However, when we set a maximum sentence length for BERT, any document or text segment longer than that limit will be truncated, which means that additional features that appear beyond the truncation point are not being taken into account. This could explain why adding all the features does not necessarily lead to better performance in our experiments. We suggest experimenting with different maximum sentence lengths (above 300) or using sliding windows, in the future.

Table 5.16: BERTimbau results using different feature combinations.

Features	Accuracy	Precision	Recall	F1-Score
description	0.62	0.50	0.47	0.45
description, title	0.66	0.55	0.50	0.49
description, title, keywords	0.65	0.54	0.48	0.48
description, title, events	0.67	0.53	0.50	0.50
description, title, entities	0.65	0.52	0.49	0.48
description, title, keywords, events	0.65	0.53	0.50	0.50
description, title, keywords, entities	0.66	0.54	0.50	0.49
description, title, events, entities	0.66	0.53	0.50	0.50
description, title, keywords, events, entities	0.66	0.53	0.49	0.50

5.5.3 Optimization Strategies

Since we have already performed hyperparameter tuning, at this stage we applied strategies to address class imbalance when training BERT with features “description”, “title”, and “events”. We assigned different weights to each class based on the class distribution in the training data. The idea is to give higher weight to underrepresented classes and lower weight to overrepresented ones. After applying class weights, the accuracy dropped to from 0.67 to 0.62, which suggests that the class weights caused the model to make more errors in classification. However, accuracy is not the most suitable metric for evaluating model performance on imbalanced datasets. The precision decreased from 0.53 to 0.52, which indicates that the model maintained a precision level similar to what it was before. The recall increased from 0.50 to 0.54, which suggests that the class weights helped the model better identify actual positive instances. Finally, the F1-Score increased from 0.50 to 0.52, which indicates that the model’s ability to balance precision and recall improved with the application of class weights.

Table 5.17: BERT Optimization.

Strategy	**Accuracy**	**F1-Score**
Base Model	0.67	0.50
Class Weighting + Base Model	0.62	0.52

5.5.4 Evaluation on Test Set

The best BERT model based on the performance on the validation set was the one trained with “description”, “title”, and “events” features, and class weighting. We evaluated its performance on a test set and achieved an Accuracy of 0.62, Precision of 0.50, Recall of 0.55, and F1-Score of 0.51. The confusion matrix is displayed in Figure 5.18, and the class-wise metrics for each complaint class are in Figure 5.19.

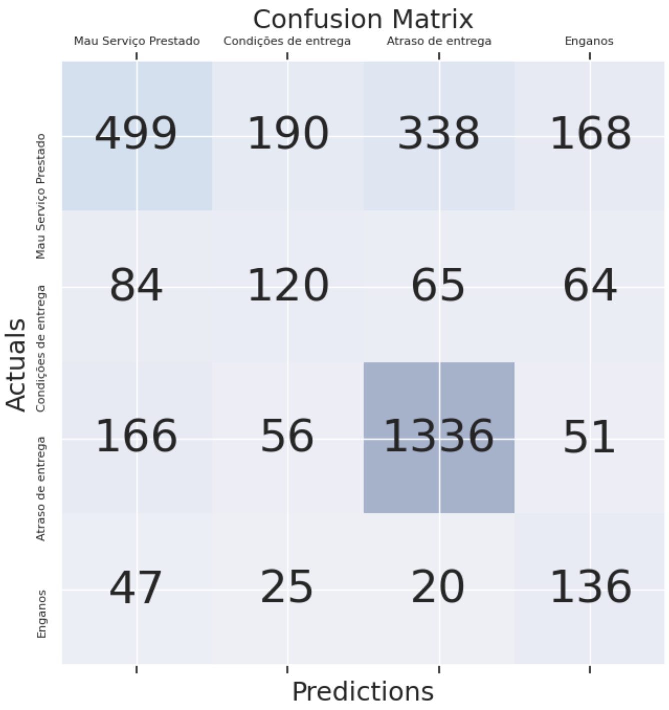


Figure 5.18: Confusion Matrix. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

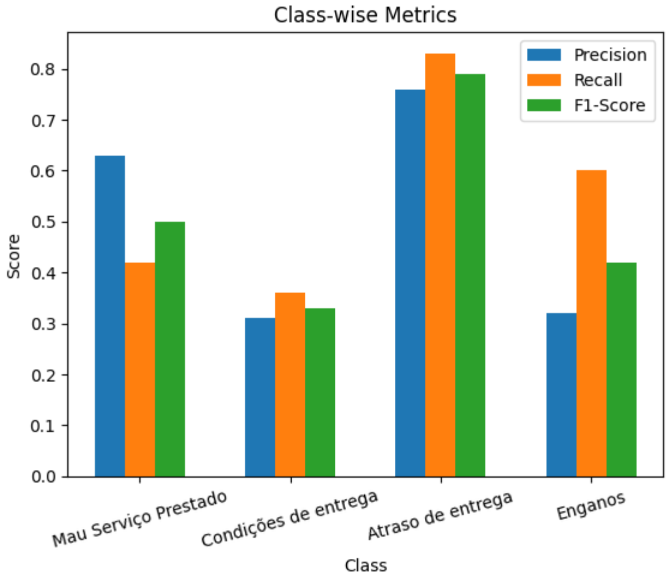


Figure 5.19: Metrics by Class. Labels: *"Condições de entrega"* - "Delivery Conditions", *"Atraso de entrega* - "Delay in Delivery", *"Enganos"* - "Mistakes", *"Mau Serviço Prestado"* - "Poor Service Provided".

The model has low precision for class *"Condições de entrega"* and class *"Enganos"*, which means that when it predicts these classes, it often makes mistakes by including instances from other classes. For class *“Condições de entrega”*, the model also has a relatively low recall, indicating that it misses a significant number of actual class *“Condições de entrega”* instances. The model has a relatively low recall for class *“Mau serviço prestado”*, meaning that it misses a significant number of actual class *“Mau serviço prestado”* instances. Nonetheless, the model performs relatively well on class *“Atraso de entrega*, with high precision and recall, indicating that it correctly identifies this class in most cases.

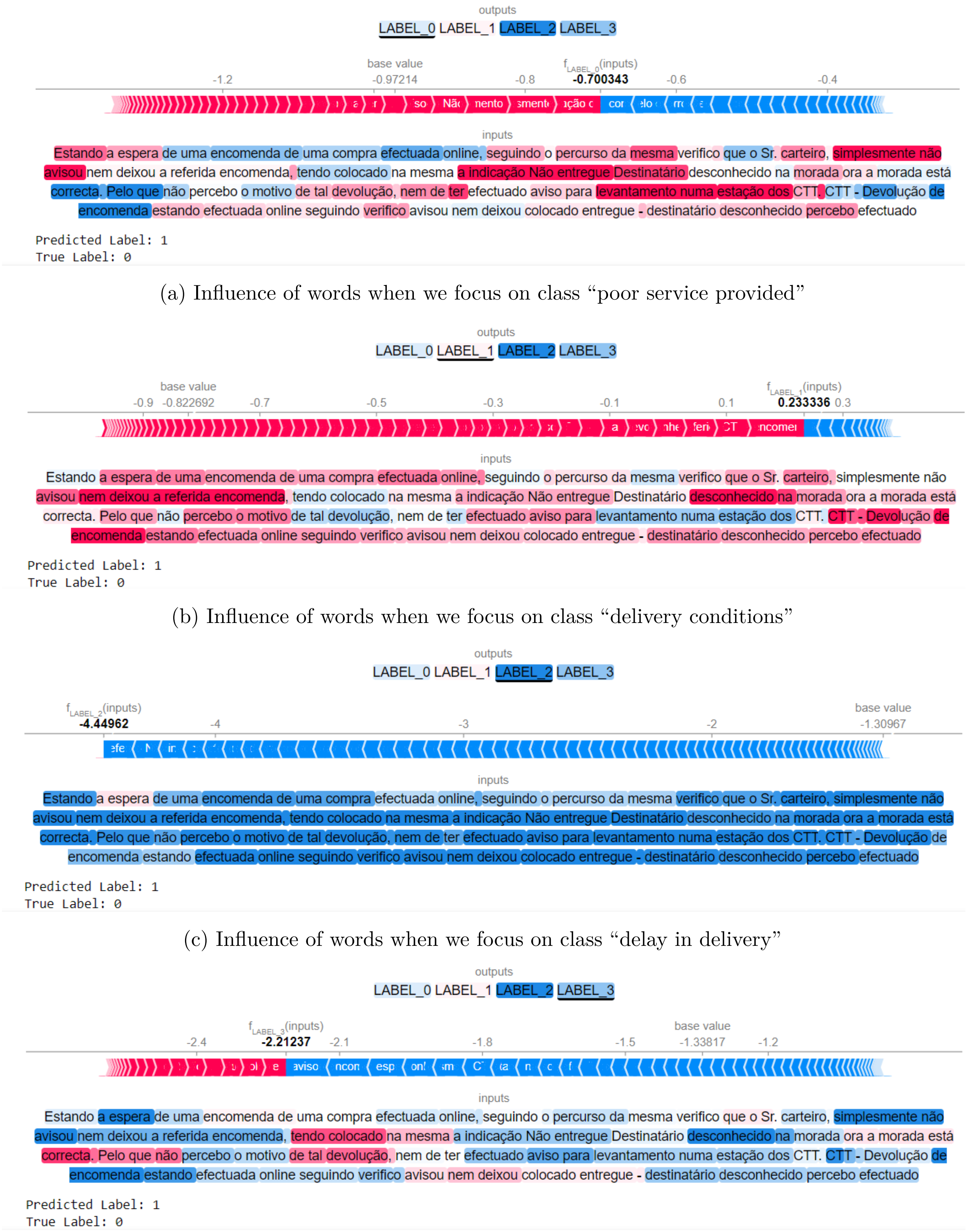
5.5.5 Model and Results Interpretation

We analyzed a sample of misclassified texts, in order to understand the model’s behavior. To do so, we used SHAP values and SHAP text plots. SHAP values represent the contribution of each feature to a model’s prediction for a specific instance. SHAP text plots are a type of visualization used to interpret the output of SHAP values in the context of text data. SHAP text plots use color to represent the magnitude and direction of the impact of each feature on the prediction. if a feature is shown in red for a specific class, it means that this feature contributes positively to the probability of that class. In other words, if we increase the value of that feature, it tends to increase the predicted probability of that class. Conversely, if a feature is shown in blue for a specific class, it means that this feature contributes negatively to the probability of that class. Increasing the value of the blue feature tends to decrease the predicted probability of that class. Features that have more impact on the score are located closer to the dividing boundary between red and blue, and the size of that impact is represented by the size of the bar.

We can visualize in Figure 5.20 the feature “contributions” towards individual classes for a specific instance, obtained using the “shap.plots.text” function of the Python library SHAP[1]. When we change the output class (LABEL_O, LABEL_1, LABEL_2, LABEL_3) at the top, corresponding to “poor service provided”, “delivery conditions”, “delay in delivery” and “mistakes”, respectively, the explanation will focus on that category. For example, in Figure 5.20a, the red features like *“simplesmente não avisou”* (“simply did not notify”), *“a indicação Não entregue Destinatário”* (“the indication Not delivered to the recipient”) and *“levantamento numa estação dos CTT”* (“pickup at a CTT station”) contribute positively towards “poor service provided” class, while the blue features *“efectuada”* (“made”) and *“Devolução de encomenda* (“return of the order”) push the prediction towards other classes. In Figure 5.20b, most features of the text drive the prediction towards class “delivery conditions”, especially *“avisou nem deixou a referida encomenda”* (“did not notify nor leave the mentioned order”), *“desconhecido”* (“unknown”) and *“CTT- Devolução de encomenda”* (“CTT - Return of the order”). In Figure 5.20c, most features push the predictions towards classes other than class “delay in delivery”, indicating that the probability of this text being predicted as “delay in delivery” is low. In fact, the text was classified as “delivery conditions”, but the true label is “poor service provided”. The text mentions that the postal worker did not provide any notification, did not deliver the package, and marked it as “Not delivered - Recipient unknown” despite the correct address. Additionally, there is mention of a notice for pickup at a CTT station, indicating issues with the

[1] https://shap.readthedocs.io/en/latest/

delivery process or conditions. These actions can indeed be interpreted as both “poor service” and “delivery conditions”.



Estando a espera de uma encomenda de uma compra efectuada online, seguindo o percurso da mesma verifico que o Sr. carteiro, simplesmente não avisou nem deixou a referida encomenda, tendo colocado na mesma a indicação Não entregue Destinatário desconhecido na morada ora a morada está correcta. Pelo que não percebo o motivo de tal devolução, nem de ter efectuado aviso para levantamento numa estação dos CTT. CTT - Devolução de encomenda estando efectuada online seguindo verifico avisou nem deixou colocado entregue - destinatário desconhecido percebo efectuado

```
Predicted Label: 1
True Label: 0
```

(a) Influence of words when we focus on class “poor service provided”

Estando a espera de uma encomenda de uma compra efectuada online, seguindo o percurso da mesma verifico que o Sr. carteiro, simplesmente não avisou nem deixou a referida encomenda, tendo colocado na mesma a indicação Não entregue Destinatário desconhecido na morada ora a morada está correcta. Pelo que não percebo o motivo de tal devolução, nem de ter efectuado aviso para levantamento numa estação dos CTT. CTT - Devolução de encomenda estando efectuada online seguindo verifico avisou nem deixou colocado entregue - destinatário desconhecido percebo efectuado

```
Predicted Label: 1
True Label: 0
```

(b) Influence of words when we focus on class “delivery conditions”

Estando a espera de uma encomenda de uma compra efectuada online, seguindo o percurso da mesma verifico que o Sr. carteiro, simplesmente não avisou nem deixou a referida encomenda, tendo colocado na mesma a indicação Não entregue Destinatário desconhecido na morada ora a morada está correcta. Pelo que não percebo o motivo de tal devolução, nem de ter efectuado aviso para levantamento numa estação dos CTT. CTT - Devolução de encomenda estando efectuada online seguindo verifico avisou nem deixou colocado entregue - destinatário desconhecido percebo efectuado

```
Predicted Label: 1
True Label: 0
```

(c) Influence of words when we focus on class “delay in delivery”

Estando a espera de uma encomenda de uma compra efectuada online, seguindo o percurso da mesma verifico que o Sr. carteiro, simplesmente não avisou nem deixou a referida encomenda, tendo colocado na mesma a indicação Não entregue Destinatário desconhecido na morada ora a morada está correcta. Pelo que não percebo o motivo de tal devolução, nem de ter efectuado aviso para levantamento numa estação dos CTT. CTT - Devolução de encomenda estando efectuada online seguindo verifico avisou nem deixou colocado entregue - destinatário desconhecido percebo efectuado

```
Predicted Label: 1
True Label: 0
```

(d) Influence of words when we focus on class “mistakes”

Figure 5.20: Single instance text plot.

We examined plots like these for a sample of misclassified instances, but for readability issues, we will include more of these examples in the Appendix B. Our analysis has revealed recurring trends that are consistent with the patterns observed in the previous sections, namely:

1. confusion between “delivery conditions” with “delivery delay” often arises when there are issues involving the delivery person claiming that the recipient was not at home and failing to leave a delivery notice.

2. confusion between “poor service provided” and “delivery conditions” when there are issues related to delivering to the wrong person or address. This confusion also arises when the delivery person falsely claims that the recipient was not at home, even when they were, or when customers pay for a more expensive service like registered mail but cannot track their packages.

3. confusion between “poor service provided” and “delay in delivery” when there are issues in customs, or when customers pay for a more expensive shipping service expecting it to be faster and more reliable, but their packages do not reach the destination as expected. This confusion also arises when there is a lack of mail delivery on a particular street.

4. confusion between “delivery conditions”/ “poor service provided” with “mistakes” when the issue involves delivering to the wrong person or address, again.

In summary, the observed issues and patterns of misclassification in the BERT-based analysis are similar to those encountered in traditional machine learning models utilizing TF-IDF features or word embeddings.

5.5.6 Summary

Table 5.18: Results Summary.

Model	**Features**	**Accuracy**	**F1-Score**
BoW + Naive Bayes (Baseline Model)	description	0.63	0.40
TF-IDF + Linear SVC	description	0.63	0.43
TF-IDF + Linear SVC	description, title	0.64	0.45
TF-IDF + Linear SVC + Class Weighting + HT	description, title	0.63	0.50
Word2Vec + XGBoost	description	0.63	0.40
Word2Vec + XGBoost	description, title	0.64	0.42
Word2Vec + Class Weighting + XGBoost + HT	description, title	0.62	0.49
GloVe + Linear SVC	description	0.64	0.40
GloVe + Linear SVC	description, title	0.65	0.42
GloVe + Class Weighting + Linear SVC	description, title	0.62	0.49
BERTimbau + HT	description	0.62	0.45
BERTimbau + HT	description, title, events	**0.67**	0.50
BERTimbau + HT + Class Weighting	description, title, events	0.62	**0.52**

In summary, the results suggest that incorporating additional features, such as “events”, can enhance the performance of the BERTimbau model for our text classification task. The use of class weighting contributes to improved model performance once again.

Moreover, the observed issues and patterns of misclassification in the BERT-based analysis are similar to those encountered in traditional machine learning models utilizing TF-IDF features or word embeddings. This could be a sign of ambiguous text or labels, or noisy data.

5.6 Results Summary

Table 5.19: Best Results Summary.

Model	**Features**	**Accuracy**	**F1-Score**
BoW + Naive Bayes (Baseline Model)	description	**0.63**	0.40
TF-IDF + Linear SVC + Class Weighting + HT	description, title	**0.63**	0.50
Word2Vec + Class Weighting + XGBoost + HT	description, title	0.62	0.49
GloVe + Class Weighting + Linear SVC	description, title	0.62	0.49
BERTimbau + HT + Class Weighting	description, title, events	0.62	**0.52**

We began our analysis by evaluating the performance of various machine learning models on our text classification task. The primary metrics we considered were accuracy and F1-score. Our baseline model, which used Bag of Words (BoW) and Naive Bayes, achieved an accuracy of 0.63 and an F1-score of 0.40. Subsequently, we explored more advanced techniques.

The TF-IDF representation, combined with a Linear Support Vector Classifier (Linear SVC), class weighting, and hyperparameter tuning (HT), also achieved an accuracy of 0.63 but improved the F1-score to 0.50. This result indicates that using TF-IDF as a feature representation and a linear classifier can enhance classification performance.

Employing Word2Vec embeddings, class weighting, XGBoost, and hyperparameter tuning yielded an accuracy of 0.62 and an F1-score of 0.49. While the accuracy slightly decreased compared to the previous model, the F1-score remained competitive, demonstrating the effectiveness of Word2Vec embeddings in text classification.

Using GloVe embeddings, class weighting, and a Linear SVC classifier achieved similar results with an accuracy of 0.62 and an F1-score of 0.49. This suggests that both GloVe and Word2Vec embeddings led to similar results.

These results indicate that the best results for each feature encoding technique tested were achieved using only “description” and “title” features and that adding additional features such as named entities, keywords, and events did not help the performance of machine learning models.

Our most advanced model, BERTimbau, with hyperparameter tuning and class weighting, achieved an accuracy of 0.62 and an impressive F1-score of 0.52. BERT-based models have shown their superiority in various NLP tasks, and our results confirm their effectiveness in text classification. Moreover, the inclusion of "events" helped the performance of the BERT model.

In the next chapter, we will analyze these results, and compare them with previous work in the field of Customer Complaint Classification. We will also try to answer the research questions, go over any limitations, and talk about where future research can go from here.

Chapter 6

Conclusions

In this final chapter, we provide a summary of our research steps, offer responses to the research questions that have guided our investigation, discuss the limitations faced during the development of this study, as well as possible future directions to follow and improve upon our work.

6.1 Research Summary

This research aimed to address several questions related to the application of machine learning algorithms in the context of customer complaint classification. We sought to identify the most effective algorithms for text classification, compare traditional machine learning approaches (KNN, Decision Trees, Random Forest, Gradient Boosting, XGBoost, Naive Bayes, Linear SVC) with deep learning models (BERT), explore the impact of various feature encoding techniques (TF-IDF, Word2Vec, GloVe), and investigate the potential of integrating keyword extraction, event extraction, and named entity recognition to enhance classification accuracy. This research was carried out in response to the growing need for effective customer support and complaint-resolution strategies in a digital and fast-paced world.

6.2 Main Findings

In this section, we summarize the main findings of the study, emphasizing the key results from each research question.

Research Question 1: What machine learning algorithms are most effective when applied to text classification problems?

Our experiments revealed that traditional machine learning algorithms such as Linear Support Vector Classifier (Linear SVC) and XGBoost are the most effective for text classification problems, especially when combined with techniques like class weighting and hyperparameter tuning. In our study, our SVM model achieved the highest F1-score of 50%, although it should be noted that our dataset was imbalanced and we used class weighting to tackle this issue.

Research Question 2: How does the performance of traditional machine learning algorithms compare to deep learning models in classifying customer complaints?

In this specific classification task, deep learning models, like BERTimbau, outperform traditional machine learning algorithms. BERTimbau achieved the highest F1-score of 0.52. However, traditional models, especially when combined with appropriate feature representations like TF-IDF or GloVe, are close to BERT-based models. Deep learning models, especially those like BERT which have a large number of parameters, often require a substantial amount of data to perform well. Our dataset for customer complaints had only 22,430 instances, and it was extremely imbalanced, so this could be why BERT did not perform even better. In contrast, LinearSVC with TF-IDF is a simpler model that can work well with smaller datasets. In fact, Lin et al. [22] demonstrate that applying pre-trained models with fixed epochs might not always lead to satisfactory results, and using linear classifiers on bag-of-words features can provide useful insights and benchmarks for text classification. The authors present experimental results comparing the performance of various linear methods (one-vs-rest, thresholding, and cost-sensitive, that use linear SVM as the binary classifier) and BERT-based models on different text classification datasets. Linear SVMs achieve competitive or even better performance than BERT-based models on some datasets. Another advantage of SVMs is that their training time and model size are significantly smaller than BERT, making them more efficient choices.

Research Question 3: What is the impact of using different feature encoding techniques, such as TF-IDF, word embeddings, or contextualized representations, on the performance of customer complaint classification?

TF-IDF appears to be a solid choice for feature encoding, especially when combined with Linear SVC. It consistently achieves competitive performance. Word2Vec and GloVe embeddings, while still effective, don't outperform TF-IDF in this specific context. They might be worth exploring further, potentially with more fine-tuning, different models, or concatenation instead of averaging. BERT-based contextualized representations, like BERTimbau, can achieve competitive performance, particularly when combined with additional features. In summary, the decision of the feature encoding technique comes down to the computational and time resources available and the dataset size.

Research Question 4: Can the integration of keyword extraction using YAKE!, event extraction using text2story, and named entities improve the accuracy of customer complaint classification in Portuguese?

The integration of keyword extraction using YAKE!, event extraction using text2story, and named entity recognition did not substantially enhance the performance of customer complaint classification in Portuguese, especially when using TF-IDF or word embeddings as encoding techniques. BERT, on the other hand, benefited the most with the inclusion of additional features. BERT, as a contextualized model, can capture contextual information from the text. Moreover, during the experimentation phase, we encountered memory limitations when attempting to increase the maximum sentence length beyond 300 tokens due to out-of-memory errors. We believe that further increasing the maximum sentence length could potentially enhance the model's performance, as the model would be able to consider more features.

Research Question 5: How can the findings from the exploratory analysis and text classification guide the development of more accurate labels for customer complaints, taking into account the identified patterns or themes in the data?

The exploratory analysis and text classification pipeline revealed that certain classes are frequently confused with one another. For example, "delay in delivery" is often misclassified as "poor service provided". In fact, complaints labeled as "poor service provided" encompass a wide range of problems, such as lost shipments, failure to meet delivery promises (e.g., when the customer pays extra for expedited delivery), customs problems, bad tracking system, lack of street service, and others. Simultaneously, all of these issues cause a delay in delivery since the customer doesn't receive the package on time or at all.

Furthermore, the "delivery conditions" class is frequently confused with the "poor service provided" class. For example, common situations in "delivery conditions" involve instances where the delivery person neglects to leave delivery notices, does not make an effort to ring the doorbell or complete the delivery process, delivers packages or letters to the incorrect recipient or address, or when items arrive damaged or in compromised condition. At the same time, all of these situations reflect a failure to provide the expected level of service quality as well.

Finally, the "mistakes" class includes situations where the packages or letters are sent to the wrong person or address, which could also cause a delay in the delivery. To create more precise and representative labels, we recommend:

1. **Revising and clarifying the definitions or descriptions of each complaint category**. For example, specify what constitutes a "Mistake" complaint versus a "Poor Service Provided" complaint, so that the customer knows what label to choose. This can help reduce confusion and misclassification.

2. **Creating more specific categories or subcategories**. Given that certain complaints can include multiple categories, creating new categories or subcategories will allow a more detailed categorization. For example, adding the category "Delivery Delays Due to Customs" to address cases where both service quality and delivery time are concerns.

3. **Implementing a multi-label classification** approach that allows complaints to be associated with multiple labels if they address more than one issue.

Based on our analysis, we can offer recommendations for improving various aspects of the business, including customer service, delivery processes, and the overall customer experience.

1. For resolving issues within the class “poor service provided”, the business should prioritize more expensive services like expedited delivery, invest in better tracking and monitoring systems, and keep customers informed about the progress of their shipments while they are in customs.
2. For resolving issues within class “delivery conditions”, the business should improve packaging quality and handling, train employees, and use technology for tracking and accountability.
3. For resolving issues within class “delay in delivery”, the business should optimize delivery routes.
4. For resolving issues within class “mistakes”, the business should request ID during package delivery for valuable items or sensitive documents.

6.3 Limitations

While this research has added some insights to the field of Portuguese customer complaint classification, it is essential to acknowledge its limitations:

- **Data Imbalance**: Some classes were underrepresented, potentially leading to a biased model towards the majority class.
- **Data Quality**: Some of the complaints included misspellings, grammatical errors, and irrelevant information, such as rants, colloquial phrases, or long narratives, which made it challenging to extract meaningful features for classification.
- **Annotation Quality**: The labels that we used were chosen by the customer at the time of the submission, which can introduce annotation bias and affect the quality of the training data.
- **Resources Constraints**: Hyperparameter tuning of ML models is time-consuming. The training, fine-tuning, and hyperparameter tuning of BERT models is computationally expensive, particularly when conducted within resource-constrained environments, such as Google Colab. For these reasons, we were not able to conduct a more exhaustive hyperparameter tuning strategy or explore alternative BERT architectures.

6.4 Future Work

Many strategies can be taken to improve and proceed with the work done in this thesis, such as:

- **Data Augmentation**: For text data, there are techniques like synonym replacement, paraphrasing, back-translation, or using generative language models (e.g., BERT) to generate text. This helps to increase the diversity of the text data while preserving its meaning.
- **New Labeling Strategy**: Use more sophisticated labeling methods, such as multilabel approaches, to account for instances with multiple types of complaints. Implement hierarchical classification techniques to capture different levels of issues within the same complaint.
- **Include Domain Knowledge**: Involve domain experts to provide more accurate and comprehensive labeling, reducing the uncertainty associated with assigning labels to complex instances.

Chapter 7

Appendix A

This Appendix contains more detailed information about the results of the experiments mentioned on sections 5.2, 5.3.1 and 5.4.1.

Table 7.1: Validation Baseline results using BoW Vectorizer on Description.

Classifier	Accuracy	Precision	Recall	F1-Score	Nr of Features
Naive Bayes	0.63	0.51	0.40	0.40	18785

Table 7.2: Best TF-IDF Configuration for Different Classifiers.

Classifier	max_df	min_df	ngram_range	Best F1-score
k-Nearest Neighbors	0.7	0.01	(1, 2)	0.3866
Decision Tree	0.7	0.02	(1, 2)	0.3566
Random Forest	0.7	0.02	(1, 3)	0.3563
Gradient Boosting	0.8	0.01	(1, 2)	0.4096
XGBoost	0.8	0.01	(1, 2)	0.4278
Naive Bayes	0.8	0.01	(1, 3)	0.3475
Linear SVC	0.8	0.01	(1, 2)	0.4322

Table 7.3: Validation results using TF-IDF Vectorizer on Description (Nr of Features = 1,191).

Classifier	Accuracy	Precision	Recall	F1-Score
k-Nearest Neighbors	0.57	0.43	0.38	0.39
Decision Tree	0.50	0.35	0.35	0.35
Random Forest	0.64	0.60	0.38	0.36
Gradient Boosting	0.64	0.53	0.41	0.41
XGBoost	0.63	0.53	0.42	0.43
Naive Bayes	0.63	0.47	0.37	0.35
Linear SVC	0.63	0.49	0.43	0.43

Table 7.4: Validation results using TF-IDF Vectorizer on Description and Title (Nr of Features = 1,240).

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-Nearest Neighbors	0.58	0.46	0.40	0.41
Decision Tree	0.51	0.37	0.36	0.36
Random Forest	0.65	0.61	0.39	0.38
Gradient Boosting	0.65	0.55	0.43	0.43
XGBoost	0.64	0.53	0.44	0.45
Naive Bayes	0.64	0.47	0.39	0.37
Linear SVC	0.64	0.50	0.44	0.45

Table 7.5: Validation results using Word2Vec embeddings on Description.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-Nearest Neighbors	0.50	0.39	0.33	0.32
Decision Tree	0.46	0.32	0.32	0.32
Random Forest	0.61	0.53	0.36	0.33
Gradient Boosting	0.63	0.49	0.40	0.40
XGBoost	0.63	0.50	0.40	0.39
Naive Bayes	0.59	0.79	0.34	0.31
Linear SVC	0.64	0.51	0.40	0.38

Table 7.6: Validation results using Word2Vec embeddings on Description and Title.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-Nearest Neighbors	0.53	0.42	0.35	0.34
Decision Tree	0.47	0.33	0.33	0.33
Random Forest	0.62	0.60	0.37	0.34
Gradient Boosting	0.64	0.51	0.42	0.42
XGBoost	0.64	0.53	0.42	0.42
Naive Bayes	0.59	0.54	0.35	0.31
Linear SVC	0.65	0.55	0.41	0.41

Table 7.7: Validation results using GloVe embeddings on Description.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-Nearest Neighbors	0.50	0.39	0.33	0.32
Decision Tree	0.45	0.32	0.32	0.32
Random Forest	0.61	0.57	0.36	0.33
Gradient Boosting	0.63	0.48	0.40	0.39
XGBoost	0.63	0.51	0.40	0.39
Naive Bayes	0.57	0.78	0.33	0.30
Linear SVC	0.64	0.52	0.41	0.40

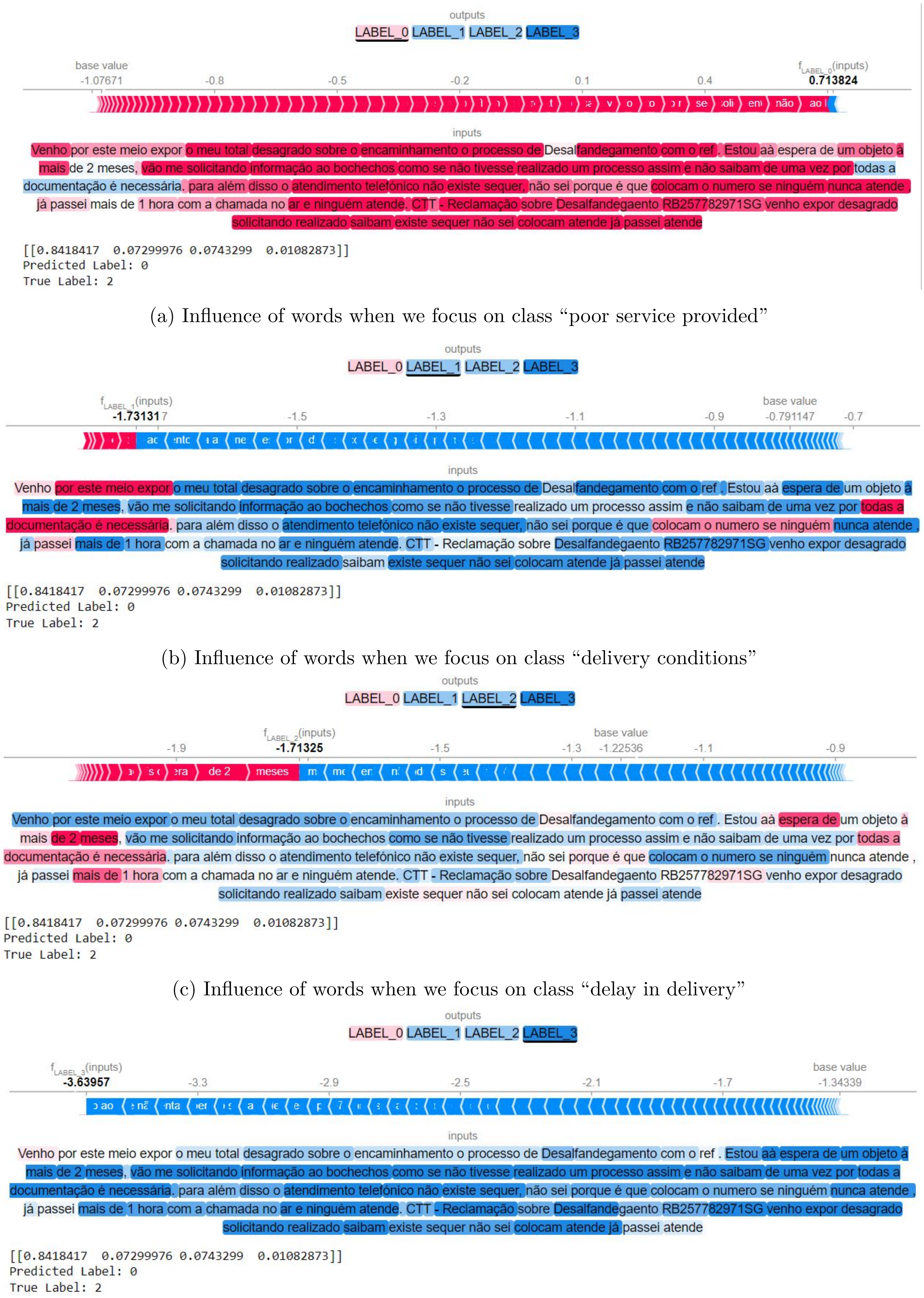
Table 7.8: Validation results using GloVe embeddings on Description and Title.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-Nearest Neighbors	0.52	0.40	0.34	0.33
Decision Tree	0.46	0.32	0.32	0.32
Random Forest	0.61	0.64	0.36	0.34
Gradient Boosting	0.63	0.51	0.41	0.40
XGBoost	0.64	0.52	0.41	0.40
Naive Bayes	0.58	0.62	0.34	0.31
Linear SVC	0.65	0.55	0.42	0.42

Chapter 8

Appendix B

This Appendix contains some examples of BERT's predictions analysis using SHAP mentioned on chapter 5.5.5.



inputs

Venho por este meio expor o meu total desagrado sobre o encaminhamento o processo de Desalfandegamento com o ref . Estou aà espera de um objeto à mais de 2 meses, vão me solicitando informação ao bochechos como se não tivesse realizado um processo assim e não saibam de uma vez por todas a documentação é necessária. para além disso o atendimento telefónico não existe sequer, não sei porque é que colocam o numero se ninguém nunca atende , já passei mais de 1 hora com a chamada no ar e ninguém atende. CTT - Reclamação sobre Desalfandegaento RB257782971SG venho expor desagrado solicitando realizado saibam existe sequer não sei colocam atende já passei atende

```
[[0.8418417  0.07299976 0.0743299   0.01082873]]
Predicted Label: 0
True Label: 2
```

(a) Influence of words when we focus on class “poor service provided”

inputs

Venho por este meio expor o meu total desagrado sobre o encaminhamento o processo de Desalfandegamento com o ref . Estou aà espera de um objeto à mais de 2 meses, vão me solicitando informação ao bochechos como se não tivesse realizado um processo assim e não saibam de uma vez por todas a documentação é necessária. para além disso o atendimento telefónico não existe sequer, não sei porque é que colocam o numero se ninguém nunca atende , já passei mais de 1 hora com a chamada no ar e ninguém atende. CTT - Reclamação sobre Desalfandegaento RB257782971SG venho expor desagrado solicitando realizado saibam existe sequer não sei colocam atende já passei atende

```
[[0.8418417  0.07299976 0.0743299   0.01082873]]
Predicted Label: 0
True Label: 2
```

(b) Influence of words when we focus on class “delivery conditions”

inputs

Venho por este meio expor o meu total desagrado sobre o encaminhamento o processo de Desalfandegamento com o ref . Estou aà espera de um objeto à mais de 2 meses, vão me solicitando informação ao bochechos como se não tivesse realizado um processo assim e não saibam de uma vez por todas a documentação é necessária. para além disso o atendimento telefónico não existe sequer, não sei porque é que colocam o numero se ninguém nunca atende , já passei mais de 1 hora com a chamada no ar e ninguém atende. CTT - Reclamação sobre Desalfandegaento RB257782971SG venho expor desagrado solicitando realizado saibam existe sequer não sei colocam atende já passei atende

```
[[0.8418417  0.07299976 0.0743299   0.01082873]]
Predicted Label: 0
True Label: 2
```

(c) Influence of words when we focus on class “delay in delivery”

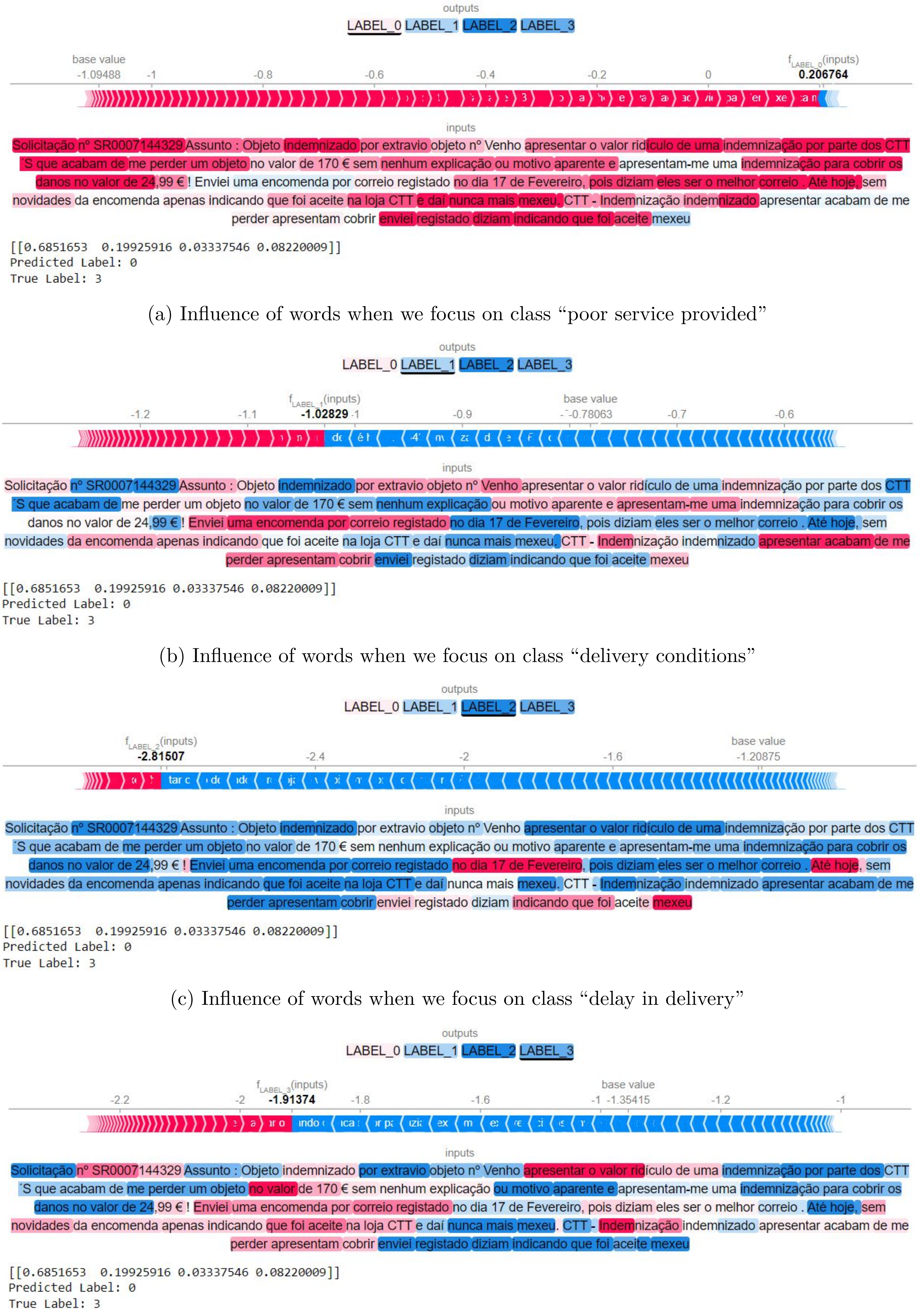
inputs

Venho por este meio expor o meu total desagrado sobre o encaminhamento o processo de Desalfandegamento com o ref . Estou aà espera de um objeto à mais de 2 meses, vão me solicitando informação ao bochechos como se não tivesse realizado um processo assim e não saibam de uma vez por todas a documentação é necessária. para além disso o atendimento telefónico não existe sequer, não sei porque é que colocam o numero se ninguém nunca atende , já passei mais de 1 hora com a chamada no ar e ninguém atende. CTT - Reclamação sobre Desalfandegaento RB257782971SG venho expor desagrado solicitando realizado saibam existe sequer não sei colocam atende já passei atende

```
[[0.8418417  0.07299976 0.0743299   0.01082873]]
Predicted Label: 0
True Label: 2
```

(d) Influence of words when we focus on class “mistakes”

Figure 8.1: Single instance text plot of a complaint labeled as “delay in delivery” but classified by BERT as “poor service provided”. Translation: “I hereby express my total dissatisfaction with the handling of the customs clearance process with reference . I have been waiting for an item for over 2 months, and they keep requesting information repeatedly, as if I had not already completed such a process, and they don’t seem to know what documentation is required once and for all. Furthermore, there is no telephone customer service available; I don’t know why they provide a phone number if nobody ever answers it. I have spent more than 1 hour with the call on hold, and nobody answers. CTT - Complaint about Customs Clearance RB257782971SG, I express my dissatisfaction and request that they be aware that customer service does not even exist. I don’t know why they provide a phone number if nobody ever answers it.”.



Solicitação nº SR0007144329 Assunto : Objeto indemnizado por extravio objeto nº Venho apresentar o valor ridículo de uma indemnização por parte dos CTT ´S que acabam de me perder um objeto no valor de 170 € sem nenhum explicação ou motivo aparente e apresentam-me uma indemnização para cobrir os danos no valor de 24,99 € ! Enviei uma encomenda por correio registado no dia 17 de Fevereiro, pois diziam eles ser o melhor correio . Até hoje, sem novidades da encomenda apenas indicando que foi aceite na loja CTT e daí nunca mais mexeu. CTT - Indemnização indemnizado apresentar acabam de me perder apresentam cobrir enviei registado diziam indicando que foi aceite mexeu

```
[[0.6851653  0.19925916 0.03337546 0.08220009]]
Predicted Label: 0
True Label: 3
```

(a) Influence of words when we focus on class “poor service provided”

Solicitação nº SR0007144329 Assunto : Objeto indemnizado por extravio objeto nº Venho apresentar o valor ridículo de uma indemnização por parte dos CTT ´S que acabam de me perder um objeto no valor de 170 € sem nenhum explicação ou motivo aparente e apresentam-me uma indemnização para cobrir os danos no valor de 24,99 € ! Enviei uma encomenda por correio registado no dia 17 de Fevereiro, pois diziam eles ser o melhor correio . Até hoje, sem novidades da encomenda apenas indicando que foi aceite na loja CTT e daí nunca mais mexeu. CTT - Indemnização indemnizado apresentar acabam de me perder apresentam cobrir enviei registado diziam indicando que foi aceite mexeu

```
[[0.6851653  0.19925916 0.03337546 0.08220009]]
Predicted Label: 0
True Label: 3
```

(b) Influence of words when we focus on class “delivery conditions”

Solicitação nº SR0007144329 Assunto : Objeto indemnizado por extravio objeto nº Venho apresentar o valor ridículo de uma indemnização por parte dos CTT ´S que acabam de me perder um objeto no valor de 170 € sem nenhum explicação ou motivo aparente e apresentam-me uma indemnização para cobrir os danos no valor de 24,99 € ! Enviei uma encomenda por correio registado no dia 17 de Fevereiro, pois diziam eles ser o melhor correio . Até hoje, sem novidades da encomenda apenas indicando que foi aceite na loja CTT e daí nunca mais mexeu. CTT - Indemnização indemnizado apresentar acabam de me perder apresentam cobrir enviei registado diziam indicando que foi aceite mexeu

```
[[0.6851653  0.19925916 0.03337546 0.08220009]]
Predicted Label: 0
True Label: 3
```

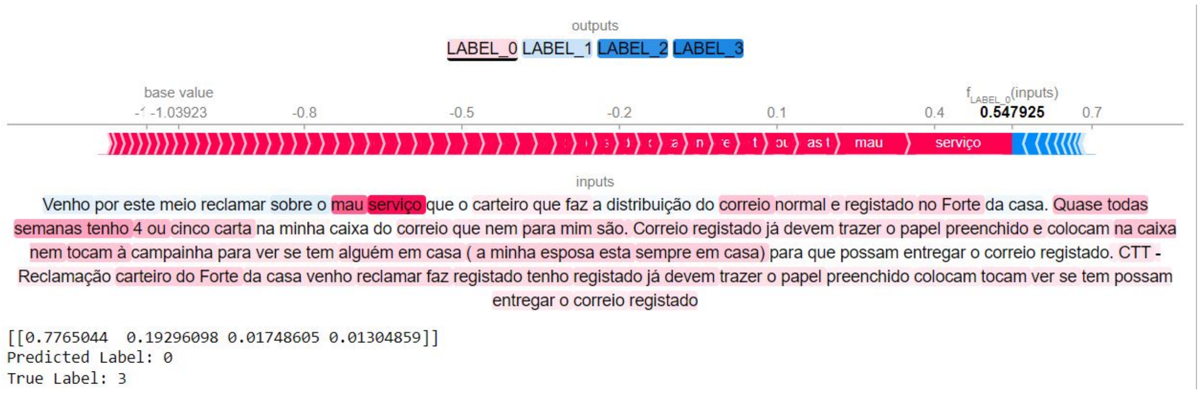
(c) Influence of words when we focus on class “delay in delivery”

Solicitação nº SR0007144329 Assunto : Objeto indemnizado por extravio objeto nº Venho apresentar o valor ridículo de uma indemnização por parte dos CTT ´S que acabam de me perder um objeto no valor de 170 € sem nenhum explicação ou motivo aparente e apresentam-me uma indemnização para cobrir os danos no valor de 24,99 € ! Enviei uma encomenda por correio registado no dia 17 de Fevereiro, pois diziam eles ser o melhor correio . Até hoje, sem novidades da encomenda apenas indicando que foi aceite na loja CTT e daí nunca mais mexeu. CTT - Indemnização indemnizado apresentar acabam de me perder apresentam cobrir enviei registado diziam indicando que foi aceite mexeu

```
[[0.6851653  0.19925916 0.03337546 0.08220009]]
Predicted Label: 0
True Label: 3
```

(d) Influence of words when we focus on class “mistakes”

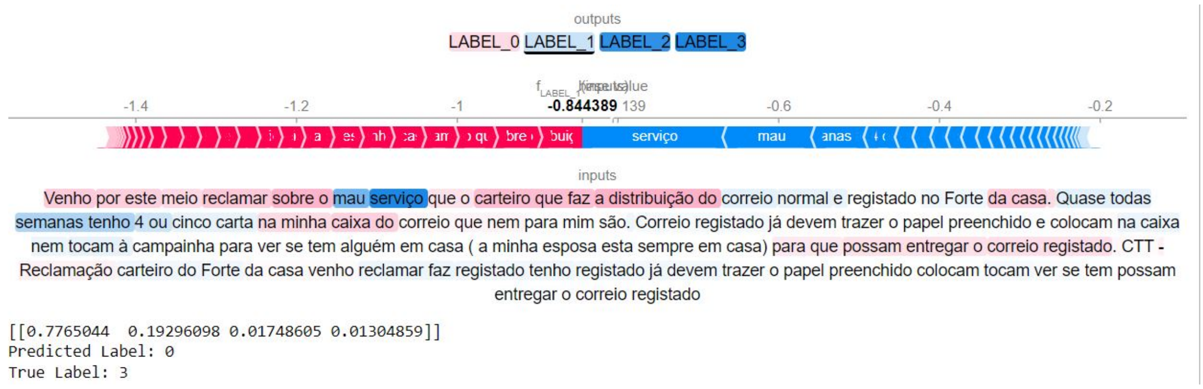
Figure 8.2: Single instance text plot of a complaint labeled as “mistakes” but classified by BERT as “poor service provided”. Translation: “Request No. SR0007144329 Subject: Compensation for Lost Item Item No. I would like to express my dissatisfaction with the ridiculously low compensation offered by CTT (Portuguese Postal Service) for losing an item worth €170 without any explanation or apparent reason. They are offering me compensation of only €24.99 to cover the damages! I sent a registered mail package on February 17th because they claimed it was the best mail service. To this day, there have been no updates on the package, except for it being accepted at the CTT store, and it hasn’t moved since. CTT - Compensation, indemnified, present, loss, offer, cover, sent, registered, claimed, indicating, accepted, hasn’t moved.”.



Venho por este meio reclamar sobre o mau serviço que o carteiro que faz a distribuição do correio normal e registado no Forte da casa. Quase todas semanas tenho 4 ou cinco carta na minha caixa do correio que nem para mim são. Correio registado já devem trazer o papel preenchido e colocam na caixa nem tocam à campainha para ver se tem alguém em casa (a minha esposa esta sempre em casa) para que possam entregar o correio registado. CTT - Reclamação carteiro do Forte da casa venho reclamar faz registado tenho registado já devem trazer o papel preenchido colocam tocam ver se tem possam entregar o correio registado

```
[[0.7765044  0.19296098 0.01748605 0.01304859]]
Predicted Label: 0
True Label: 3
```

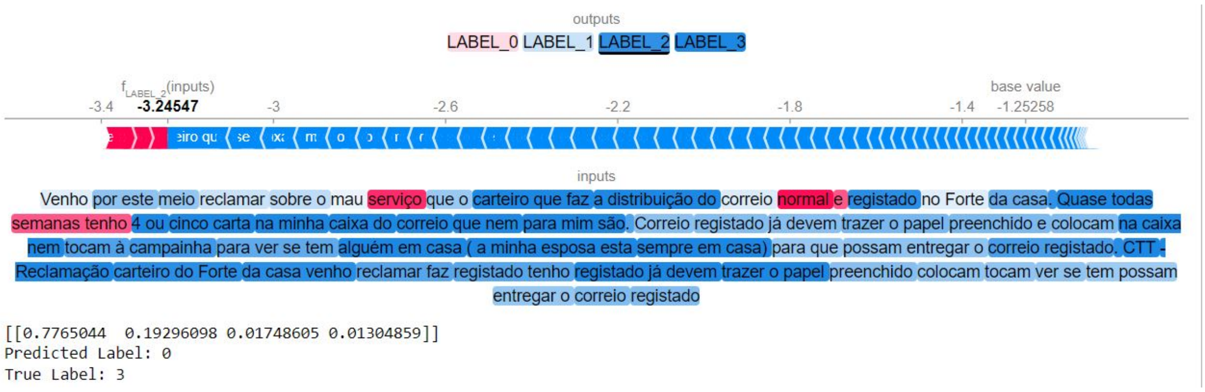
(a) Influence of words when we focus on class "poor service provided"



Venho por este meio reclamar sobre o mau serviço que o carteiro que faz a distribuição do correio normal e registado no Forte da casa. Quase todas semanas tenho 4 ou cinco carta na minha caixa do correio que nem para mim são. Correio registado já devem trazer o papel preenchido e colocam na caixa nem tocam à campainha para ver se tem alguém em casa (a minha esposa esta sempre em casa) para que possam entregar o correio registado. CTT - Reclamação carteiro do Forte da casa venho reclamar faz registado tenho registado já devem trazer o papel preenchido colocam tocam ver se tem possam entregar o correio registado

```
[[0.7765044  0.19296098 0.01748605 0.01304859]]
Predicted Label: 0
True Label: 3
```

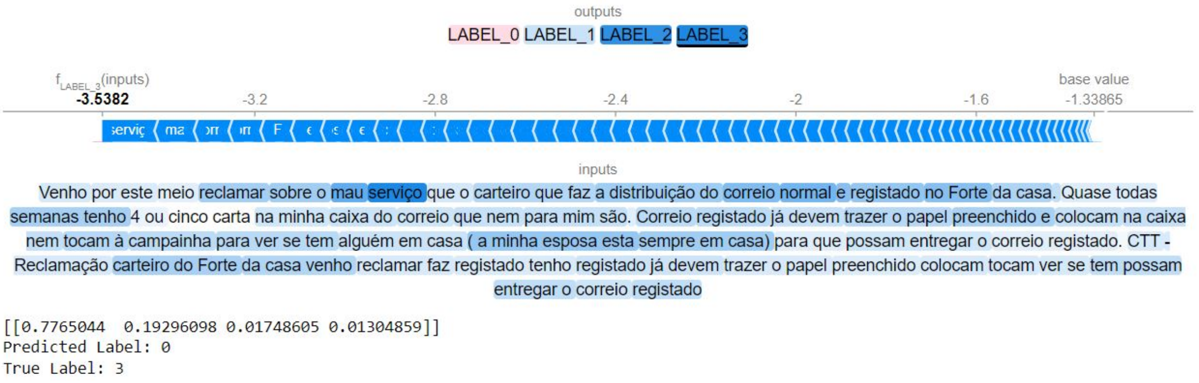
(b) Influence of words when we focus on class "delivery conditions"



Venho por este meio reclamar sobre o mau serviço que o carteiro que faz a distribuição do correio normal e registado no Forte da casa. Quase todas semanas tenho 4 ou cinco carta na minha caixa do correio que nem para mim são. Correio registado já devem trazer o papel preenchido e colocam na caixa nem tocam à campainha para ver se tem alguém em casa (a minha esposa esta sempre em casa) para que possam entregar o correio registado. CTT - Reclamação carteiro do Forte da casa venho reclamar faz registado tenho registado já devem trazer o papel preenchido colocam tocam ver se tem possam entregar o correio registado

```
[[0.7765044  0.19296098 0.01748605 0.01304859]]
Predicted Label: 0
True Label: 3
```

(c) Influence of words when we focus on class "delay in delivery"

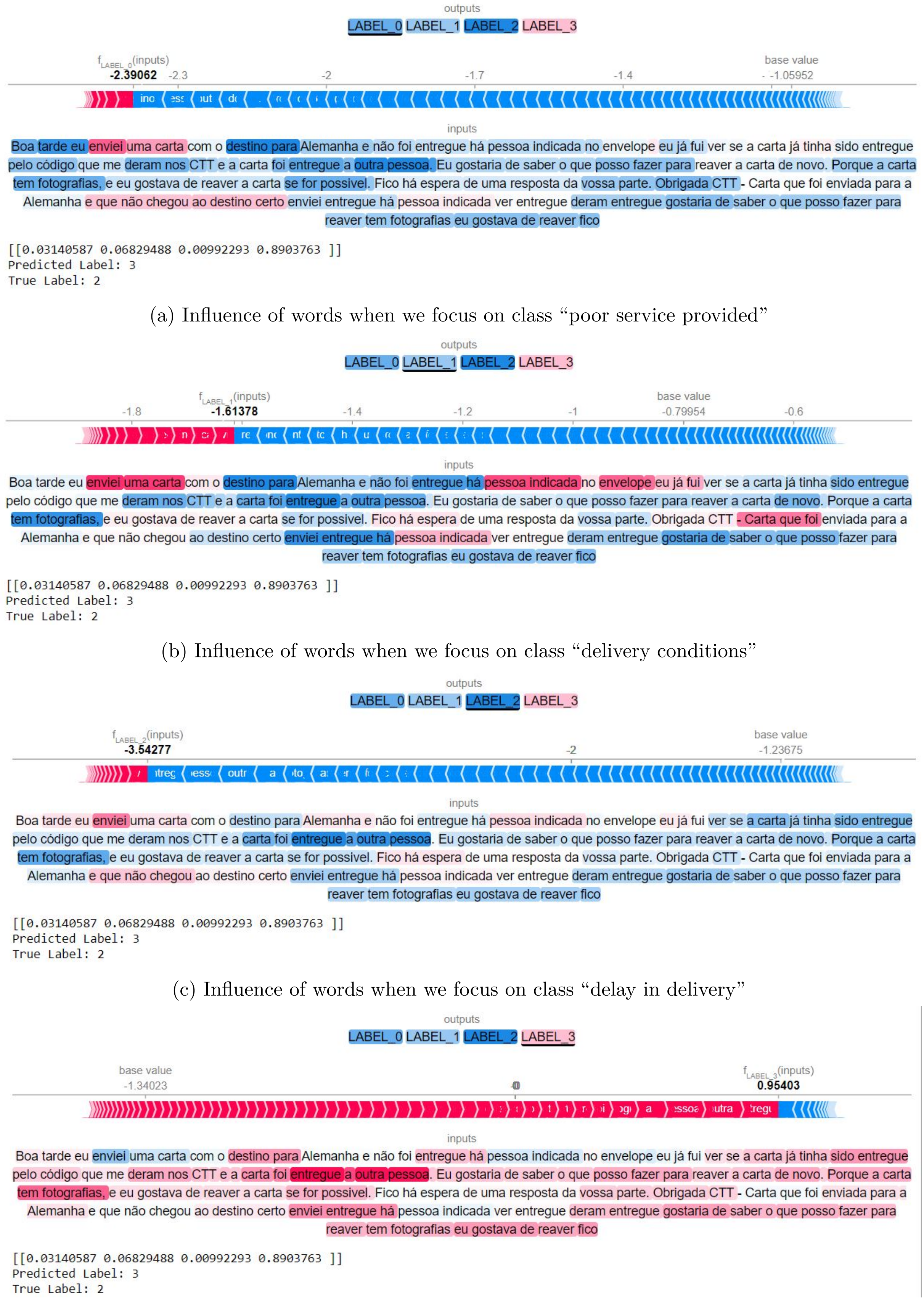


Venho por este meio reclamar sobre o mau serviço que o carteiro que faz a distribuição do correio normal e registado no Forte da casa. Quase todas semanas tenho 4 ou cinco carta na minha caixa do correio que nem para mim são. Correio registado já devem trazer o papel preenchido e colocam na caixa nem tocam à campainha para ver se tem alguém em casa (a minha esposa esta sempre em casa) para que possam entregar o correio registado. CTT - Reclamação carteiro do Forte da casa venho reclamar faz registado tenho registado já devem trazer o papel preenchido colocam tocam ver se tem possam entregar o correio registado

```
[[0.7765044  0.19296098 0.01748605 0.01304859]]
Predicted Label: 0
True Label: 3
```

(d) Influence of words when we focus on class "mistakes"

Figure 8.3: Single instance text plot of a complaint labeled as "mistakes" but classified by BERT as "poor service provided". Translation: "I hereby wish to complain about the poor service provided by the mail carrier who handles the delivery of regular and registered mail in Forte da Casa. Almost every week, I receive 4 or 5 letters in my mailbox that are not even addressed to me. Registered mail should already have the necessary paperwork filled out, and they simply place it in the mailbox without ringing the doorbell to check if anyone is home (my wife is always at home) so that they can deliver the registered mail. CTT - Complaint about the mail carrier in Forte da Casa, I come to complain that they handle registered mail, I have registered mail, they should already bring the paperwork filled out, they put it in the mailbox, they don't ring the bell to see if they can deliver the registered mail.".



Boa tarde eu enviei uma carta com o destino para Alemanha e não foi entregue há pessoa indicada no envelope eu já fui ver se a carta já tinha sido entregue pelo código que me deram nos CTT e a carta foi entregue a outra pessoa. Eu gostaria de saber o que posso fazer para reaver a carta de novo. Porque a carta tem fotografias, e eu gostava de reaver a carta se for possivel. Fico há espera de uma resposta da vossa parte. Obrigada CTT - Carta que foi enviada para a Alemanha e que não chegou ao destino certo enviei entregue há pessoa indicada ver entregue deram entregue gostaria de saber o que posso fazer para reaver tem fotografias eu gostava de reaver fico

```
[[0.03140587 0.06829488 0.00992293 0.8903763 ]]
Predicted Label: 3
True Label: 2
```

(a) Influence of words when we focus on class “poor service provided”

Boa tarde eu enviei uma carta com o destino para Alemanha e não foi entregue há pessoa indicada no envelope eu já fui ver se a carta já tinha sido entregue pelo código que me deram nos CTT e a carta foi entregue a outra pessoa. Eu gostaria de saber o que posso fazer para reaver a carta de novo. Porque a carta tem fotografias, e eu gostava de reaver a carta se for possivel. Fico há espera de uma resposta da vossa parte. Obrigada CTT - Carta que foi enviada para a Alemanha e que não chegou ao destino certo enviei entregue há pessoa indicada ver entregue deram entregue gostaria de saber o que posso fazer para reaver tem fotografias eu gostava de reaver fico

```
[[0.03140587 0.06829488 0.00992293 0.8903763 ]]
Predicted Label: 3
True Label: 2
```

(b) Influence of words when we focus on class “delivery conditions”

Boa tarde eu enviei uma carta com o destino para Alemanha e não foi entregue há pessoa indicada no envelope eu já fui ver se a carta já tinha sido entregue pelo código que me deram nos CTT e a carta foi entregue a outra pessoa. Eu gostaria de saber o que posso fazer para reaver a carta de novo. Porque a carta tem fotografias, e eu gostava de reaver a carta se for possivel. Fico há espera de uma resposta da vossa parte. Obrigada CTT - Carta que foi enviada para a Alemanha e que não chegou ao destino certo enviei entregue há pessoa indicada ver entregue deram entregue gostaria de saber o que posso fazer para reaver tem fotografias eu gostava de reaver fico

```
[[0.03140587 0.06829488 0.00992293 0.8903763 ]]
Predicted Label: 3
True Label: 2
```

(c) Influence of words when we focus on class “delay in delivery”

Boa tarde eu enviei uma carta com o destino para Alemanha e não foi entregue há pessoa indicada no envelope eu já fui ver se a carta já tinha sido entregue pelo código que me deram nos CTT e a carta foi entregue a outra pessoa. Eu gostaria de saber o que posso fazer para reaver a carta de novo. Porque a carta tem fotografias, e eu gostava de reaver a carta se for possivel. Fico há espera de uma resposta da vossa parte. Obrigada CTT - Carta que foi enviada para a Alemanha e que não chegou ao destino certo enviei entregue há pessoa indicada ver entregue deram entregue gostaria de saber o que posso fazer para reaver tem fotografias eu gostava de reaver fico

```
[[0.03140587 0.06829488 0.00992293 0.8903763 ]]
Predicted Label: 3
True Label: 2
```

(d) Influence of words when we focus on class “mistakes”

Figure 8.4: Single instance text plot of a complaint labeled as “delay in delivery” but classified by BERT as “mistakes”. Translation: “Good afternoon, I sent a letter destined for Germany, but it was not delivered to the intended person indicated on the envelope. I checked to see if the letter had been delivered using the code provided by CTT, and it turns out that the letter was delivered to someone else. I would like to know what I can do to retrieve the letter. The letter contains photographs, and I would like to get it back if possible. I am waiting for a response from your side. Thank you. CTT - Letter sent to Germany that did not reach the correct destination, I sent it, delivered to the intended person indicated, checked, given, delivered, would like to know what I can do to retrieve it, contains photographs, I would like to retrieve it, waiting.”.

Bibliography

[1] Charu C. Aggarwal. *Data Mining - The Textbook*. Springer, 2015. ISBN: 978-3-319-14142-8. 2.2.1

[2] M. Anand, Kishan Bhushan Sahay, Mohammed Altaf Ahmed, Daniyar Sultan, Radha Raman Chandan, and Bharat Singh. Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. *Theoretical Computer Science*, 2022. ISSN: 0304-3975. doi:https://doi.org/10.1016/j.tcs.2022.06.020. 3.2

[3] Lucas Vinicius Avanço and Maria das Graças Volpe Nunes. Lexicon-based sentiment analysis for reviews of products in brazilian portuguese. In *2014 Brazilian Conference on Intelligent Systems*, pages 277–281, 2014. doi:10.1109/BRACIS.2014.57. 3.2

[4] Micaela Bangerter, Giuseppe Fenza, Mariacristina Gallo, Vincenzo Loia, Alberto Volpe, Carmen De Maio, and Claudio Stanzione. Unisa at SemEval-2023 task 3: A SHAP-based method for propaganda detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 885–891, Toronto, Canada, July 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.semeval-1.122. 3.4, 3.4

[5] Francisco Caldeira, Luís Nunes, and Ricardo Ribeiro. Classification of Public Administration Complaints. In João Cordeiro, Maria João Pereira, Nuno F. Rodrigues, and Sebastião Pais, editors, *11th Symposium on Languages, Applications and Technologies (SLATE 2022)*, volume 104 of *Open Access Series in Informatics (OASIcs)*, pages 9:1–9:12, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN: 978-3-95977-245-7. doi:10.4230/OASIcs.SLATE.2022.9. 3.2, 3.2.1, 3.2.2, 3.2, 3.5

[6] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020. ISSN: 0020-0255. doi:https://doi.org/10.1016/j.ins.2019.09.013. 2.3

[7] Matheus Henrique Cardoso, Anita Maria da Rocha Fernandes, Giovani Marin, Valderi Reis Quietinho Leithardt, and Paul Crocker. Comparison between different approaches to sentiment analysis in the context of the portuguese language. In *2021 16th*

Iberian Conference on Information Systems and Technologies (CISTI), pages 1–6, 2021. doi:10.23919/CISTI52073.2021.9476501. 3.2

[8] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. 2.2.1

[9] Douglas Cirqueira, Márcia Fontes Pinheiro, Antonio Jacob, Fábio Lobato, and Ádamo Santana. A literature review in preprocessing for sentiment analysis for brazilian portuguese social media. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 746–749, 2018. doi:10.1109/WI.2018.00008. 3.2

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. doi:10.48550/ARXIV.1810.04805. 2.2.2

[11] Fernando Leandro Dos Santos and Marcelo Ladeira. The role of text pre-processing in opinion mining on a social media language dataset. In *2014 Brazilian Conference on Intelligent Systems*, pages 50–54, 2014. doi:10.1109/BRACIS.2014.20. 3.2

[12] Heba Elfardy, Manisha Srivastava, Wei Xiao, Jared Kramer, and Tarun Agarwal. Bingo at IJCNLP-2017 task 4: Augmenting data using machine translation for cross-linguistic customer feedback classification. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 59–66, Taipei, Taiwan, December 2017. Asian Federation of Natural Language Processing. 3.1, 3.1

[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9(61):1871–1874, 2008. 4.4.1

[14] Anna Glazkova. A comparison of synthetic oversampling methods for multi-class text classification. *CoRR*, abs/2008.04636, 2020. 3.3, 3.3

[15] Nathan Hartmann, Erick R. Fonseca, Christopher Shulby, Marcos V. Treviso, Jéssica S. Rodrigues, and Sandra M. Aluísio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *CoRR*, abs/1708.06025, 2017. 3.2.2, 4.3.2

[16] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. ISBN: 9780387848846. 2.2.1

[17] Daniel Jurafsky and James H. Martin. *Speech and Language Processing, Third Edition draft*. 2023. 2.1.3, 2.2.2, 2.3

[18] John D. Kelleher, Brian MacNamee, and Aoife D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press, Cambridge, MA, 2015. ISBN: 978-0-262-02944-5. 4.4.3, 4.4.3

[19] Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown. Text classification algorithms: A survey. *Information*, 10(4):150, apr 2019. doi:10.3390/info10040150. 2.1.3

[20] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019. ISSN: 2078-2489. doi:10.3390/info10040150. 2.1.1, 2.1.2

[21] Vivian Lai, Jon Z. Cai, and Chenhao Tan. Many faces of feature importance: Comparing built-in and post-hoc feature importance in text classification. *CoRR*, abs/1910.08534, 2019. 3.4, 3.4

[22] Yu-Chen Lin, Si-An Chen, Jie-Jyun Liu, and Chih-Jen Lin. Linear classifier: An often-forgotten baseline for text classification, 2023. 6.2

[23] Henrique Lopes-Cardoso, Tomás Freitas Osório, Luís Vilar Barbosa, Gil Rocha, Luís Paulo Reis, João Pedro Machado, and Ana Maria Oliveira. Robust complaint processing in portuguese. *Information*, 12(12), 2021. ISSN: 2078-2489. doi:10.3390/info12120525. 3.2, 3.2.1, 3.2.2, 3.2, 3.5

[24] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN: 9781510860964. 2.4

[25] Ricards Marcinkevics and Julia E. Vogt. Interpretability and explainability: A machine learning zoo mini-tour. *ArXiv*, abs/2012.01805, 2020. 2.4

[26] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. 2.1.3

[27] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN: 9780262018029 0262018020. 2.2

[28] Annalisa Occhipinti, Louis Rogers, and Claudio Angione. A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*, 201: 117193, 2022. ISSN: 0957-4174. doi:https://doi.org/10.1016/j.eswa.2022.117193. 3.4, 3.4

[29] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi:10.3115/v1/D14-1162. 2.1.4

[30] Heng Rathpisey and Teguh Bharata Adji. Handling imbalance issue in hate speech classification using sampling-based methods. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, pages 193–198, 2019. doi:10.1109/ICSITech46713.2019.8987500. 3.3, 3.3

[31] Jeffrey S. Saltz. Crisp-dm for data science: Strengths, weaknesses and potential next steps. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2337–2344, 2021. doi:10.1109/BigData52589.2021.9671634. 2.5

[32] Sara Silva, Rubén Pereira, and Ricardo Ribeiro. Machine learning in incident categorization automation. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, 2018. doi:10.23919/CISTI.2018.8399244. 3.1, 3.1

[33] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Bertimbau: Pretrained bert models for brazilian portuguese. In Ricardo Cerri and Ronaldo C. Prati, editors, *Intelligent Systems*, pages 403–417, Cham, 2020. Springer International Publishing. ISBN: 978-3-030-61377-8. 4.4.1

[34] Frederico Dias Souza and João Baptista de Oliveira e Souza Filho. Sentiment analysis on brazilian portuguese user reviews. In *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6, 2021. doi:10.1109/LA-CCI48322.2021.9769838. 3.2

[35] Frederico Dias Souza and João Baptista de Oliveira e Souza Filho. Embedding generation for text classification of brazilian portuguese user reviews: from bag-of-words to transformers. *Neural Computing and Applications*, dec 2022. doi:10.1007/s00521-022-08068-6. 3.2

[36] Seba Susan and Amitesh Kumar. The balancing trick: Optimized sampling of imbalanced datasets—a brief survey of the recent state of the art. *Engineering Reports*, 3(4):e12298, 2021. doi:https://doi.org/10.1002/eng2.12298. 3.3

[37] Jafar Tanha, Yousef Abdi, Negin Samadi, Nazila Razzaghi-Asl, and Mohammad Asadpour. Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7:70, 09 2020. doi:10.1186/s40537-020-00349-y. 3.3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. doi:10.48550/ARXIV.1706.03762. 2.2.2

[39] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. doi:10.1109/MCI.2018.2840738. 1.1