Machine Learning and Deep Learning Algorithms to Correct and Classify Product Reviews in a Marketplace

Licinio Daniel Gomes Carvalho

Dissertação de Mestrado

Orientador na FEUP: Prof. Luís Gonçalo Rodrigues Reis Figueira

Mestrado em Engenharia e Gestão Industrial

2022-07-11

Resumo

A rápida expansão do número de plataformas no mercado actual do comércio electrónico torna o mesmo extremamente competitivo. Como tal, as empresas precisam de compreender as necessidades dos consumidores e as suas principais preocupações para desenvolver novas abordagens para tornar os seus serviços mais atractivos. Os comentários online têm um impacto significativo neste mercado porque é onde muitos consumidores procuram informação antes de realizarem uma compra. Neste caso, foi detectado que a empresa não tinha forma de categorizar por assunto os diferentes comentários e que existiam algumas discrepâncias entre a classificação por estrelas dada pelo consumidor e o texto apresentado no comentário.

Por conseguinte, seria importante compreender a diferença entre a classificação por estrelas e o elemento textual de cada comentário e analisar em que categorias relacionadas com o negócio a empresa necessita melhorar o seu desempenho.

Esta dissertação visa compreender como as técnicas consideradas estado de arte de Machine Learning e Deep Learning podem ser utilizadas para a classificação de textos. Estas técnicas serão utilizadas para desenvolver uma nova classificação textual para cada comentário e classificá-lo numa categoria relacionada com o negócio, utilizando o conjunto de dados fornecido pela empresa. Para efeitos de comparação, foram criados dois protótipos de dashboards para analisar os resultados.

A classificação de texto é uma tarefa complexa, uma vez que o texto é composto por várias ambiguidades, e especialmente, os comentários são compostos de opiniões dos consumidores. Além disso, estes comentários, na maioria das vezes, contêm elementos sobre diferentes aspectos do negócio.

As previsões para o novo rating textual foram obtidas utilizando a técnica de Supervised Learning *XGBoost* e permitiram detectar vários discrepâncias entre o actual rating e o novo rating textual desenvolvido. Para além disso, a técnica de classificação de Unsupervised Zero-Shot apresentou resultados impressionantes já que obteve uma precisão decente na classificação das diferentes revisões em seis categorias: *stock*, *produto*, *entrega*, *preço*, *descrições*, e *marketplace*. Utilizando ambos dashboards, a empresa detectou um fraco desempenho nas categorias de *stock* e *entrega*.

Abstract

The rapid expansion in the number of platforms in today's e-commerce makes the market extremely competitive. As such, companies need to understand consumers' needs and their main concerns to develop new ways of making their services more appealing. Online reviews have a significant impact in this market because it is where many consumers look for information before they make a purchase. In this case, it was detected that the company had no way to categorize the different reviews by topic (e.g. whether it is related to the product, its availability, etc.) and that there were some mismatches between the star rating given by the consumer and the text presented in the review.
Therefore, it would be important to understand the difference between the star-rating and the textual element of each review and to analyze in which business-related categories the company needs to improve its performance.
On this note, this dissertation aims to understand how state-of-the-art Machine Learning and Deep Learning techniques can be used for text classification. These techniques will be used to develop a new textual rating to each review and classify them into business-related categories using the dataset provided by the company. For comparison purposes two dashboards' prototypes were created to analyze the results.
Classifying text is a difficult task as text is composed of several ambiguities, and especially, reviews are composed of opinions of the consumers. Furthermore, these reviews, most of the time, contain elements on different aspects of the business.
The predictions of the new textual rating were obtained using the Supervised Learning XGBoost technique and allowed to detect several mismatches between the current rating and the new textual rating developed. Furthermore, the Unsupervised Zero-Shot Classification technique presented impressive results and obtained a decent accuracy in classifying the different reviews into six categories: *stock*, *product*, *delivery*, *price*, *descriptions*, and *marketplace*. Using both dashboards, the company detected underperformance in the predicted categories of *stock* and *delivery*.

Key Words: Text Classification, Machine Learning, Deep Learning, Marketplace

Agradecimentos

Ao Professor Gonçalo Figueira pelo suporte, disponibilidade e conselhos que me guiaram durante o desenvolvimento desta dissertação.
À Worten pela oportunidade de desenvolver a dissertação junto da equipa de CRO, em especial, à minha orientadora Gabriela Campos pela disponibilidade de informação e apoio.
As minhas próximas palavras são dedicadas à pessoa que tornou este percurso possível, o meu pai, Licinio Carvalho, que sempre me mostrou que com esforço e perseverança nada é impossível.
Um obrigado também a toda a minha família, em especial, à minha namorada Carina Pinto por todo o apoio e ao Eng. José Pedro pela sua disponibilidade.
Queria também agradecer a todas as pessoas que participaram na avaliação dos resultados pela sua sincera opinião e pela disponibilidade.
Finalmente, gostaria de agradecer a todos os elementos do grande grupo de amigos que sempre me acompanhou ao longo deste percurso académico.

Daniel Carvalho.

"Business is war, and your past clients and customers' great online reviews are your elite soldiers in battle"

Tom Kenemore

Contents

Acronyms and Symbols

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-term Memory units
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRO	Conversion Rate Optimization
CNN	Convolution Neural Network
CTT	Correios, Telégrafos e Telefones
DBN	Deep Belied Network
DL	Deep Learning
DNN–MHAT	Deep Neural Network Multi-Head Attention
ELMo	Embeddings from Language Models
e-WOM	Electronic Word-of-Mouth
KPI	Key Performance Indicator
LSTM	Long Short-Ter memory networks
ME	Maximum Entropy
ML	Machine Learning
MLM	Masked Language Modeling
NB	Naïve Bayes
NCR	National Cash Register Co.
NLTK	Natural Language Toolkit
PLM	Pre-Trained Language Models
POS	Part of Speech
RNN	Recurrent Neural Network
S-BERT	sentence-based BERT model
SEMMA	Sample, Explore, Modify, Model, Assess
SGPS	Equity Management Company (Sociedade Gestora de Participações Sociais)
SKU	stock-keeping unit
SOM	Self Organizing Maps
SPSS	Statistical Package for the Social Sciences
SVM	Support Vector Machine
tfidf	term frequency-inverse document frequency
XGBoost	Extreme Gradient Boosting
ZSC	Zero-Shot Classification

List of Figures

List of Tables

Chapter 1

Introduction

The growth and impact of e-commerce on our life is impressive. Endlessly scrolling online stores' platforms searching for products has become so frequent that it can almost be considered a habit. The idea of an online shopping system has been around for a long time since it was invented by Michael Aldrich, the "Godfather of e-commerce," in 1979 (Cowan, 2021). Since then, many important historical marks have been written in this field; however, one that can be classified as one of the most important was the launch of Amazon.com by Jeff Bezos in 1995, currently the most significant and largest e-commerce company.

E-commerce was defined in 1997 by Rolf Wigand as: "the seamless application of information and communication technology from its point of origin to its endpoint along the entire value chain of business processes conducted electronically and designed to enable the accomplishment of a business goal" (Wigand, 1997). This definition from R. Wigand seems complex, and it is indeed challenging to understand. Over time, many authors in the literature have tried to give a more straightforward explanation of the concept of e-commerce. In a broad sense, e-commerce can be described as: "Any form of business relationship where interaction between actors occurs through the use of Internet technologies" (Babenko and Syniavska, 2018).

The number of active e-commerce consumers and companies in this field has significantly increased over the past years. As a result, the e-commerce sector has become tremendously competitive. Nowadays, more than ever, businesses must be able to better understand their customers to provide them with relevant products and services.

Moreover, e-commerce platforms sincerely rely on the image presented to the public. That is why the different products presented to the customer significantly influence the number of sales of an online retail company. Companies should highlight in their website the products that are most relevant to consumers. One important problem e-tailers need to address is therefore to determine in which order the products should be presented to customers; this is one of the functions of the CRO department. On the same note, errors in the assignment of the review rating by consumers can impact the overall rating of a product, especially in those products with fewer reviews, as is the case of products from Marketplaces, which will cause the same products to appear lower than intended.

Furthermore, e-commerce businesses depend on several areas such as the website presentation, the description of the products, managing stock, the logistic partners used for the distribution of the products, and, if applicable, the reliability of the Marketplace sellers who sell products through the website. Depending on so many different areas, it is essential to identify the most critical ones so that they can quickly improve them. Following this idea, vast amounts of information can be extracted and analyzed from customer feedback through reviews. This extracted and adequately classified information can later be used to detect the consumer's discontent regarding a specific area.

1.1 Worten and the Sonae Group

The information presented in this section was sourced from the SONAE Group and the Worten website. The Sonae SGPS, S.A. is a multinational corporation present in 62 countries that manages a portfolio of businesses that create value in different regions. Sonae is a holding company that oversees and actively manages a consumer-focused portfolio of businesses, operated independently, grouped into several categories: retail, shopping center construction and administration, fixed and mobile telecommunications, media, and new technologies. The different businesses are operated independently.
Worten is Sonae's retail company in charge of the electronics area. Currently, Worten has 26 years of existence and is present throughout the entire Portuguese country, including the autonomous regions of Madeira and Azores and in Spain. Some important dates relevant for this dissertation are the following:

- 1996 - Inauguration of the first store in Chaves;
- 2001 - Launch of the online store www.worten.pt;
- 2009 - With the acquisition of the Boulanger outlets, Worten officially joins the Spanish market;
- 2018 - Launch of Marketplace at www.worten.pt;
- 2021 - Worten sells 17 stores in Spain. The transaction also includes the future closure of 14 stores in Spain, only keeping one physical store in Madrid and 15 stores in the Canary Islands. This transaction represents the will of Worten to focus on sales through the online channel in Spain;
- 2021 - Worten absorbed Dott.pt.

Worten started its journey with the idea of selling electronic products; however, with the launch of the marketplace, nowadays, Worten has it all, from Home & Decoration to Beauty, Health & Baby, from Big and Small Household Appliances to Sound & Image goods, from Computers to Telecommunications, Entertainment, Gaming, and Culture. Naturally, the marketplace launch

increased the number of products on Worten's website, reaching new concerns related to sales through Marketplace sellers.
The company is committed to customers by offering the best price/quality ratio. To accomplish this commitment reviewing, documenting, and improving all processes is essential; however, Worten must understand which points should be prioritized.

1.2 The company and the E-commerce market

This section has the objective of introducing the reader to the current situation of Worten in the e-commerce market, especially in the Portuguese and Spanish market, and to describe the current situation of the marketplace implemented in Worten.

1.2.1 The company in the Portuguese market

A study made by Alberto Pimenta, shows that the lockdown due to the Covid-19 pandemic has provoked a great leap in the use of e-commerce by Portuguese people. Data provided by the CTT e-commerce report shows that in 2020 the e-commerce in Portugal has seen a growth of 46,4% reaching an impressive number of 4,4bn euros in the transaction of goods (Alberto Pimenta, 2021). It was predicted that this rate of growth was going to slow down with the vaccination process and the opening of physical commerce, however, 2021 was a year of consolidation for the e-commerce market in Portugal reaching a growth near the 23%.
Analysis from the netAudience measurement system conducted by Marktest (2021) and only counting personal computer users show how Worten's website as lead the e-commerce race in Portugal until June 2021 in front on websites like Aliexpress, Amazon, Continente, Fnac and Booking. Information extracted from Algolia, platform used by Worten, shows how in November 2021, Worten saw a total number of users of 2,69 million total users and 7,92 million total searches in the Portuguese Website. In May 2022, the number reached 3,05 million and total users and 8,86 million total searches.

1.2.2 The company in the Spanish market

In the case of Spain, a study developed by IAB Spain (2021) shows that, as we have previously seen in the Portuguese case, the lockdown due to the Covid-19 pandemic has also provoked a growth in the number of users. In 2020 Spain saw growth in e-commerce around the 36% mark. Spain has also reached 76% of people between ages 16 and 70 who had already made an online purchase.
A study developed by ecommerceDB shows that for Worten, the picture is different in Spain; the website is not present in the top 10 e-commerce websites. The market is dominated by the presence of amazon.es and elcorteingles.es (EcommerceDB, 2021). As previously noted, information extracted from the Algolia platform shows how in November 2021, Worten saw a total number of

users of 378 100 total users and 723 400 total searches on the Spanish Website. In May 2022, the number reached 327 307 total users and 699 882 total searches.

1.3 The company and the rise of Marketplaces

The worldwide growth of the e-commerce market in recent years has forced companies to arouse their interest in this opportunity. However, the introduction to this new market can be highly demanding, especially for small resellers and manufacturing companies, as it can represent a significant financial investment that could result in a lack of visibility and difficulties turning the investment into profit.

Furthermore, the introduction of marketplaces has increased competition as it promotes more competitive prices and reduces the market power of well-established sellers. However, some barriers still concern the consumer, mainly in the field of infrastructures, and the fear and lack of confidence in online retailers due to online illiteracy (Fernández-Bonilla et al., 2022).

On this note, literature shows that the number of retailers offering their products through new channels, such as marketplaces or agency selling, has skyrocketed (Wei and Dong, 2022). Big retail platforms such as Amazon, Alibaba, and Jingdong have allowed independent suppliers to show and sell their products directly to the consumer through virtual storefronts (Zhang and Zhang, 2020; Zhang et al., 2020). The impact of this new channel has proven to be very important for the e-commerce industry seeing examples such as the third quarter of 2020 for Amazon, where 54% of the total revenue was made in the marketplace channel (Guo et al., 2021) or the case of JD.com, Jingdong, were more than 270 000 retailers had joined the e-commerce platform in 2019 (Shi et al., 2021).

As previously noted, Worten launched its Marketplace in 2018, and since then, the number of marketplace sellers has increased, reaching around 1500 sellers nowadays. As for the products, Worten has a total of 5,8 million SKUs published, where 41% are available for purchase, corresponding to a total of 2,4 million SKUs. The distribution of products is presented in Figure 1.1.

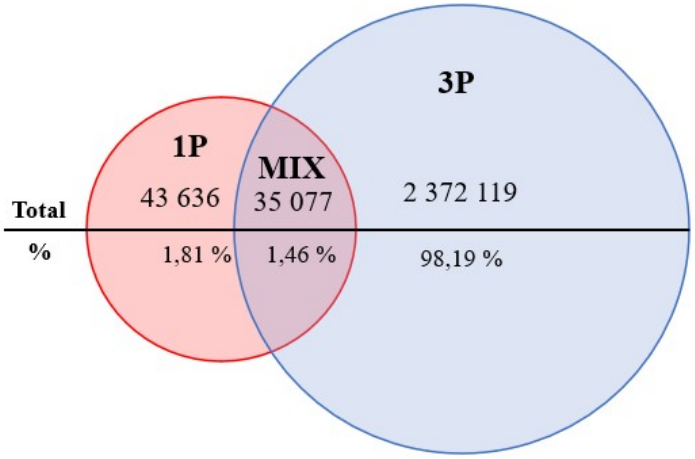


Figure 1.1: Total number of products sold by Worten vs Marketplace

The weight of the marketplace products (3P) in Worten's platform is essential as they represent more than 95% of the total products available for sale.
Moreover, the company's brand and communication director, António Fuzeta da Ponte, is confident that the marketplace brings several advantages to the consumer, one of them being the possibility of finding a wide variety of products at competitive prices and believes that the number of marketplace sellers will increase (Cristina A. Ferreira, 2021).
However, as previously noted, the reliability aspect of marketplace sellers is a critical aspect to consumers as the quality of the products and the services provided is the total responsibility of the sellers. To ensure the reliability of the Marketplace, Worten selects each seller and has the right to remove sellers with the low product quality or low quality of service from the Marketplace.

1.4 Objectives and the Methodology used

This dissertation focuses on two main challenges: i) detecting errors in current product ratings by developing a new textual rating for each review; and ii) categorizing each review according to the underlying topic for further analysis of the principal causes of consumer dissatisfaction.

1.4.1 Multi-class Classification of online reviews

The Search Software *Algolia* is responsible for ordering the products on Worten's website for both category pages and term searches. On the one hand, category pages display all products that belong to a particular parent and child *product category*. On the other hand, when the consumer searches for a specific product-related term, such as the name of the product, the search is considered a term search.
Algolia makes decisions according to several variables such as the term used to search for the product, if the search is done by search term, a binary variable that indicates if a product is in stock, the overall rating of the product, the number of times the product is added to the cart, or the number of times the product is added to a wish list.
Knowing how the platform orders the products, several machine learning (ML) and deep learning (DL) techniques of supervised learning are explored to predict a 1-to-5 rating for reviews. Creating this new *Textual Rating* allows Worten to have a way of quantifying the impact of some errors in the current and future review's rating. That consequently can influence the ordering of products within the website, making the website more faithful to the consumer's opinion, and possibly boosting sales.
Moreover, to better understand consumers' opinions and take actions to improve processes, it is necessary to analyse the consumer's feedback on the website. This can be done by analyzing consumer reviews. However, it is impossible to analyze thousands of comments on the website and extract useful information efficiently.
To increase the efficiency of further analysis, it is necessary to classify reviews into specific categories. This can allow a quicker performance analysis of each field and increase the possibility of

detecting and correcting related problems. Reviews can be related to several areas such as products' *descriptions*, *delivery*, availability of *stock*, *price*, or *marketplace* sellers.
Deep learning (DL) techniques are explored using Transformer-Based Pre-Trained Language Models (PLMs) to create a way of classifying reviews coherently according to the main points of concern early described. These models are implemented in a fully unsupervised text classification with Zero-Shot Classification (ZSC).

1.4.2 The CRISP-DM Methodology

The consolidation of *Data Mining* as an essential tool to analyze and understand the data produced in industries created the necessity to establish industry-wide standards. One of the standards is the pre-defined guiding steps for implementing the different *Data Mining* methods.
Following this need, two main methods are presented as the most relevant at this point. The CRISP-DM (Cross-Industry Standard Process for Data Mining) and the SEMMA (Sample, Explore, Modify, Model, Assess). These methods have been compared in works such as Ana Azevedo's (Azevedo and Santos, 2008) and, more recently, Christoph Schröer's (Schröer et al., 2021).
However, these two works show that the CRISP-DM methodology is more complete than the SEMMA one (Azevedo and Santos, 2008). Another advantage of this CRISP-DM method is that it is easy to interpret and structure, its reliable, and it is an industry-independent process (Schröer et al., 2021).
This methodology was developed by a partnership between DaimlerChrysler, SPSS, and NCR and can be applied in multiple industries for tasks related to data mining (Azevedo and Santos, 2008). Furthermore, as the CRISP-DM methodology is considered the most complete method this dissertation will be developed using it. Figure 1.2 presents the six different iterative steps of the CRISP-DM methodology.

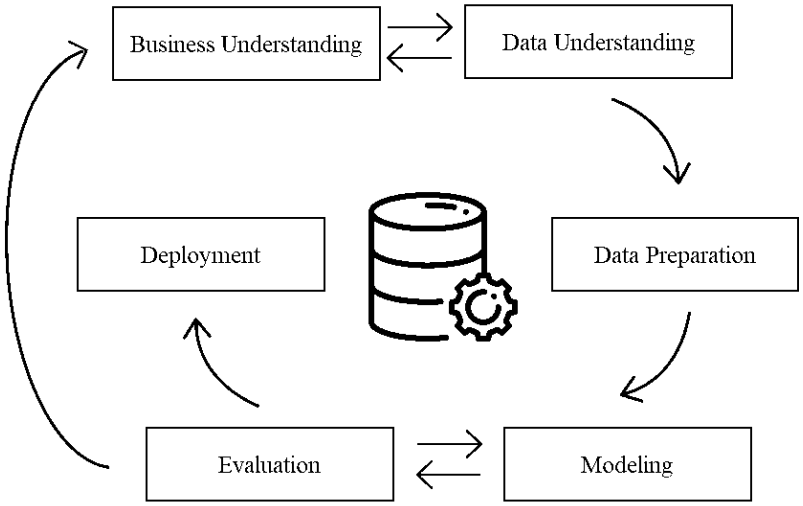


Figure 1.2: CRISP-DM steps. Adapted from Azevedo and Santos (2008)

- *Business understanding*, where the objective is to understand the needs of the company, and convert objectives and requirements into a data mining problem. This phase is presented in Sections 1.2, 1.3 and 3.1;
- *Data understanding*, where the objective is to collect the data and explore it extracting insights. This step consisted of contacting the Worten team responsible for managing the reviews on the platform, understanding what data could be extracted and which elements help develop the dissertation. This phase is presented in Section 3.2;
- *Data preparation* presents how the final dataset is constructed. In this step, the dataset is divided into a training, a test, and a validation set using the stratified sampling method. Then, as the data extracted (raw data) contained different languages, spelling errors, unknown characters, and even fake reviews composed of random keyboard spam by the consumer, the pre-processing of the data was performed. This phase is presented in Sections 3.3 and 3.4;
- *Modeling* describes the different models used for each task. In this step three methods for supervised learning and one method for unsupervised classification were selected and described in Sections 2.6, 2.7, 3.5 and 3.6;
- *Evaluation*, where each model's results and the different steps should be evaluated. This step is essential to define the best model to predict the most appropriated rating and category for each review and is developed in Sections 4.1 and 4.2;
- *Deployment*, where the information obtained should be organized and presented. In this step, covered in Section 4.3, it is mandatory to develop a dashboard that can be used to quickly analyze the predicted results.

System architecture

For the Supervised Learning methods, the Python language was used in the Jupyter Notebook in the Anaconda software that allows to simplify the package management and deployment for developing the Machine Learning and Deep Learning methods. On the other hand, for Unsupervised Classification it was used the Google Colaboratory (Colab) that is a web-product from Google Research based in the Python language that is useful to create and run arbitrary code when more computing power is required.

1.5 Structure

This section serves the purpose of describing the structure of this document.
In Chapter 2, the importance of the CRO department in the company and the importance of reviews for consumers is presented. Then, the related works in text classification for Supervised and Unsupervised methods are analyzed to define the state-of-the-art techniques that should be used in the dissertation. Finally, a more in-depth description of the proposed models implemented is presented.
Chapter 3 introduces how the company deals with reviews and the technologies currently used. Then, it presents the different datasets used for the development of this dissertation and the techniques used for the treatment of text. Furthermore, this section presents the transformations for the datasets and the different libraries and parameters defined for each model used.
Chapter 4 presents and discusses the results obtained with those models and displays a visualization obtained from the best results.
Chapter 5 presents the final conclusions of this dissertation.

Chapter 2

Literature Review

This chapter focuses on describing the principal concepts employed in this thesis. As such, this section introduces the importance of the CRO Department, the importance of the review's textual element, and the numeric rating in reviews. Furthermore, it analyzes related works developed in concepts such as *Text Mining*, *Natural Text Processing* (NLP), *Data Pre-processing*, and *Text Classification*. Finally, a more in-depth description of the proposed models for Supervised and Unsupervised Classification is presented.

2.1 The importance of the CRO department and the rating of reviews

This chapter aims to introduce the reader to the extent of marketing in the e-commerce sector, presenting the function of the Conversion Rate Optimization (CRO) and the importance of reviews in e-commerce platforms through the analysis of the existing literature.

2.1.1 Conversion Rate Optimization (CRO)

The easy accessibility to new information about products and their respective prices has provoked an increase in competition between e-commerce platforms. Alberto Pimenta (2021) describes today's consumers as: "mature and more involved with new technologies". This idea that the success of an e-commerce platform has become more demanding is supported by Perez Amaral et al. (2019) work .

The concept of a more demanding e-consumer combined with the limits of online platforms, such as not being able to have a personal interaction with the salesman or even not being able to touch and feel the overall quality of products, has a significant impact on the shopping experience. Following this idea, other variables such as *Customer Satisfaction* and *Loyalty* are crucial for the success of online retail stores. Philip Kotler, the father of marketing, was asked by Peter Drucker in an interview if the purpose of a company was to create customers; his answer was clear and consistent: "No customers, no paycheck" (Gunther, 2009).

Literature has proved that e-commerce platforms rely on several factors contributing to a better shopping experience. Rita et al. (2019) enumerates variables such as the web design, the quality and differentiation of the products offered, and the data security of the website. More recently, Griva (2022) confirms the previously presented idea and adds that an attractive layout of the web page, the easiness of payment, and the fact of having multiple payment methods also impact *Customer Satisfaction*.

If e-commerce platforms want to be competitive in today's market, it is essential to understand, learn and plan how to optimize customer experience and loyalty (Zhang et al., 2013).

That is why CRO (Conversion Rate Optimization) has become one of the leading development points for e-companies. Miikkulainen et al. (2017) define CRO as the science that studies web interfaces to optimize the conversation of casual users into actual customers. However, some authors define CRO as an art. Tim Ash, the CEO of SiteTuners, once stated: "Conversion Rate Optimization (CRO) is the art and science of getting people to act once they arrive on your website" (Josh Steimle, 2015). This idea of classifying CRO as art can derive from different procedures and techniques such as testing layouts, visual designs, and neuromarketing (Josh Steimle, 2015).

This concept profoundly relies on the conversion rate *KPI* (Key Performance Indicator) presented in Equation 2.1. It represents the percentage of visitors, in the case of e-commerce platforms, that make an action considered a conversion. Several actions can be regarded as a conversion depending on the company's objective; for e-commerce platforms, the most critical one is adding a product to the cart or making a purchase.

$$Conversion\ Rate\ (\%) = \frac{Total\ number\ of\ a\ specific\ actions}{Total\ number\ of\ visitors} \times 100 \tag{2.1}$$

Conversion rate is known to have relatively low values; as Miikkulainen et al. (2017) exhibits, they are typically 2% to 4%.

Furthermore, the Click Through Rate KPI is also important as it can be seen as how consumers express the product's attractiveness. In Equation 2.2, the number of impressions is the number of times the product is presented to the consumer, and the number of clicks represents the number of total clicks that the product obtained.

$$Click\ Through\ Rate\ (\%) = \frac{Total\ number\ of\ clicks}{Total\ number\ of\ impressions} \times 100 \tag{2.2}$$

These simple but very important metrics have become one of the main metrics that online platforms use to analyze user behavior, engagements, purchases and organize products in the platform (Saleem et al., 2019).

2.1.2 Importance of rating

The emergence of multiple marketplace platforms allows consumers to choose from a growing number of possibilities, from which product to buy to the platform where to buy it. To compete in the online retailing market, platforms should understand the necessities of a more knowledgeable

consumer.

In the literature, several studies have been focused on understanding consumers' necessities concluding that e-WOM (Electronic Word-of-Mouth) could shape the consumer's awareness and perception of a product (Geng et al., 2019; Hu et al., 2008). In other words, there is a possibility that consumers that lack the experience of using a product can be influenced by the 'voice of the consumer' about several aspects, such as quality, usability, and reliability, among others. The concept relays on the idea that the consumer can build an image of how the product is and how it will perform through e-WOM (Geng et al., 2019).

Following the previously described idea, there are three main aspects of reviews that can impact the e-consumer's perception. The first one is the review valence; this aspect is based on the mean user rating that consumers have given to a product; the second one is the volume of reviews; the number of total reviews that a product has; the third one is the variance of the reviews; in other words, the inconsistency in ratings given by consumers.

Studies have been developed to understand if these previously described aspects impact e-consumers' perception and, consequently, if there is a relation between them and the future product sales. Hu et al. (2008) explains how only the valence in movie reviews has a significant and positive impact on future earnings of box offices. Moreover, disagreeing with Hu's findings that only valence impacts sales, J. Chevalier showed by comparing books' reviews from Amazon.com and bn.com that the website with a better rating for the same product had a larger sales volume (Chevalier and Mayzlin, 2006). Supporting this idea, Kostyra et al. (2016) demonstrates that a higher number of reviews with a high rating can be an indicator of confidence to new customers having a greater probability of consumers choosing the product and reducing the impact of other product aspects such as the brand. On the other hand, a study developed using user reviews in movies by Duan et al. (2008) supports the idea that the valence does not affect the sales and only the volume of reviews significantly impacts future movie revenue.

Summing up, in literature, it is possible to see the importance of the valence and the volume of reviews in the e-consumers' perception and behavior; however, the impact of variance is neglected. Understanding what has a real impact on reviews is a crucial concept because the consumer, having defined an idea through the reviews of others, can compare different products and platforms, choosing those with a greater valence and volume of reviews. Furthermore, companies can analyze e-WOM around a product. They can use this to produce better product forecasting, use this knowledge to develop new products, and learn how to attract and retain new consumers (Geng et al., 2019).

There are two current ways of reviewing a product on e-commerce platforms; the consumer can evaluate a product with a star rating, representative of a numeric value, and a text review. Companies must understand which of these reviewing methods is more taken into account by consumers.

Star rating

Reviewing through star rating is a quantitative method to evaluate a product (Geng et al., 2019); it quantifies the consumer's overall experience with the product. It is the fastest and easiest way to rate a product where the consumer must choose a number between one and five. The value 1

represents a negative feeling, the value 3 represents a neutral feeling, and the number 5 represents an excellent experience with the product. It gives other readers an idea of the overall feeling without details about the product's features.
The star rating is, most of the time, the first impression that an e-consumer has about the product and, as the simulation developed by Moe et al. (2011) indicates, the difference in the star rating in products has an indirect effect on the product's sales. On the same note, the overall rating of a product itself can influence future star rating classifications (Moe et al., 2011).

Textual element

On the other hand, the reviewing through text is an open-ended description of the reviewer's opinion of the product (Geng et al., 2019). This reviewing method reveals a more detailed and deep thoughted product experience, and it can contain practical insights into product characteristics that offer helpful information for new buyers (Chevalier and Mayzlin, 2006).
In literature, it has been documented that this reviewing method performs an essential role in customer choices. The report conceived by Alberto Pimenta (2021) points out that the existence of comments from other buyers on a product represents one of the main reasons Portuguese consumers give for buying products online . Also, even if it is not fast and practical to read multiple reviews, Chevalier and Mayzlin (2006) reveal that e-consumers read the comments and that these textual reviews can have an emotional positive or negative contagious effect on other buyers, and that is why it represents a variable that has a significant bearing on their decision of buying a product.
Moreover, this reviewing method allows consumers to identity-relevant information such as the reviewer's reputation (Forman et al., 2008). This aspect is the key to building buyers' trust in the product and the platform when the comments contain positive, relevant information about the platform's or the product's functionality (Nikolay et al., 2011).
Finally, it is important to underline that these textual reviews can also serve as a powerful predictive tool that companies can use to predict possible consumer behavior and possible consumer demand and to explain the variation in product demand over changes in customer sentiment (Nikolay et al., 2011).

Star rating vs. Textual review

Having previously noted in this document that consumers pay attention to these two review methods and that these impact future sales, it would be interesting to understand which is the more critical element or if they work together to create a combination and if they have an influence one in the other.
Hu et al. (2006) points out a specific problem with a star rating; besides only showing an overall consumer feeling that is insufficient to capture the natural feel of a consumer, these ratings suffer from bimodality, in other words, consumers will demonstrate an extreme feeling over a product, and they will give a rating extremely high or extremely low. On the other hand, textual reviews demand more work from the possible buyer.
Even if star ratings reflect more of an overall feeling and textual reviews reflect more sentiment-

topic aspects (Geng et al., 2019), there is also a connection between the two reviewing methods. Star rating and textual reviews are not independent. In the literature, it is known that star rating influences the sentiment expressed in textual reviews and that customers might only read reviews that are classified as a high star rating if they are very interested in the product, or they might only read textual reviews that are classified as low star rating is they want to understand which characteristics are less advantageous in a specific product (Geng et al., 2019).
It would be interesting to understand if other external product attributes, such as the strength of the brand, have an impact on reviews. However, this is not discussed in this document.
Having seen the multiple aspects of reviews, a conclusion can be taken: numerical and textual reviews are essential and have an increasing influence, even if one more than another, on the sales performance of a product. E-commerce companies should dedicate a growing interest in analyzing reviews because they enable a better understanding of how the products and services are performing and can be used to identify future changes on the platform (Catelli et al., 2022).

2.2 Text mining

Today's natural use of the World Wide Web has introduced a new way for consumers to express themselves and give opinions about their experiences with a product or the service provided by the company. This opinion is considered more unbiased than in other media as it is known that the internet allows people to have a mask over their identity, allowing consumers to express their opinions without being influenced by fear, pressure, intimidation, or incentives (Raghupathi et al., 2014).
Following the idea that online retail stores are an essential medium of communication between different users where interactions can result in sharing ideas, knowledge, and experience about a product or a platform through reviews (Tripathy et al., 2015), a vast amount of textual information can be extracted and analyzed by companies.
Text mining focuses on the problem of identifying and extracting knowledge from fuzzy and unstructured data. It is a problematic field of data mining. However, unlike other areas, it focuses on the process of natural language text. It can be defined as processing extensive textual data and finding patterns, models, trends, facts, rules, or relationships in unstructured data that are not easily perceived for further analysis (Nahm and Mooney, 2002).
Text mining can be considered a multi-disciplinary field as it incorporates and combines several tools and advanced techniques of data mining and NLP, such as information extraction and information retrieval, statistical pattern learning, topic modeling, computational linguistics, and machine learning (Nahm and Mooney, 2002; Sourav et al., 2022). This document will address ML and DL techniques for text mining.
As text mining is the ability to extract information from extensive unstructured data, multiple tasks can be accomplished using text mining, such as text clustering, entity or concept extraction, taxonomies, sentiment analysis, document summarization, and entity-relation modeling (Jiawei Han, 2014). As Jiawei Han (2014) describes in his book, these tasks can be beneficial in fields like

financial data analysis, telecommunication industries, science and engineering, spam detection, marketing research, and, overall, in all areas that are related to decision making.
As can be seen in Figure 2.1, Text Mining requires several steps (Jiawei Han, 2014; Sourav et al., 2022):

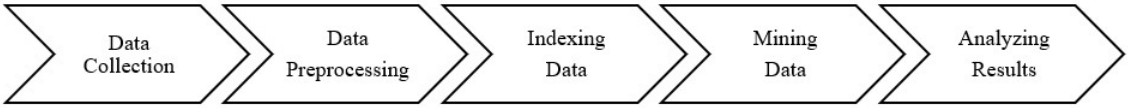


Figure 2.1: Text Mining steps

1. The first step involves extracting data from different sources such as reviews from websites, retail stores, forums, or even social networks. This information collected is unstructured and contains several components that can influence the output of the process;

2. Secondly, the natural text extracted from the different sources needs to be cleaned. This step is called Data Pre-processing; the user should detect and remove irregularities in the collected data. For this step, many NLP and Text Mining tools can be used. This is also the step where users should focus more of the time as it is a lengthy step that significantly impacts the output of the process;

3. After pre-processing the extracted data, the information should be converted into a structured format and stored so it can be quickly accessed; this step is called indexing;

4. The next step is called mining and is associated with using different techniques of computer science to discover patterns, rules, and relationships within the structured data;

5. The last step in text mining is evaluating the model and analyzing and interpreting the results.

In this dissertation, this steps were implemented using the CRISP-DM methodology described in Section 1.4.2. Moreover, it is essential to highlight, as Jiawei Han referred in his book, that getting a result in text mining that is considered to have high quality usually is related to the combination of the relevance and interestingness of the theme in analysis and not to the performance of the computational tools (Jiawei Han, 2014).

2.3 Natural Language Processing (NLP)

Companies should be aware of the importance of spontaneous reviews (Tripathy et al., 2015). However, this information that is scattered in the comments of online retail stores, blogs, and forums, is information that is not sorted or interpreted, also called raw data, and needs to be collected and analyzed. This is where NLP (Natural Language Processing) is a convenient tool.
Natural Language Processing saw its birth around the 1950s (Nadkarni et al., 2011) and is seen in literature as an area of research in computational linguistics that has the objective of exploring

how computers can be used to scientifically extract, identify, study and understand information from natural language text (Jagdale et al., 2019; Parveen et al., 2013). In other words, it can be described as the ability to use computational tools and techniques to understand human language as it is spoken.

The NLP field is difficult for investigation where different theories and techniques deal with the problem of natural language (Khurana et al., 2017). Humans can understand natural language on seven interdependent levels: phonological, morphological, lexical, syntactic, semantic, discourse, and pragmatic. These levels should be fully developed to have a complete understanding of the message. However, as described in the literature, computers can only analyze four different levels. At the word level, computers can analyze the morphology (most minor parts of a word like suffixes and prefixes) and the semantics of the words (the meaning of the word). At a sentence level, it can be analyzed the lexical of a sentence (the parts of speech) and the syntax (the structure of the sentence and the order of the words) (Parveen et al., 2013).

Major challenges in NLP

Due to the complex nature of human communication, working with natural text represents many challenges for computer science. Several literature documents expose that NLP is deeply influenced by the ambiguity related to the nature of words or sentences (Pathak and Thankachan, 2012) and metaphors (Barnden, 2008).

According to Pathak and Thankachan (2012), some ambiguities related to text can be divided into several categories:

- *Lexical ambiguity* - A simple word can have a different meaning depending on the context of the document;
- *Syntactic ambiguity* - The same sentence can be put together and be parsed in different ways;
- *Referential ambiguity* - The use of pronouns can cause difficulty in selecting a unique referent for a linguistic expression;
- *Pragmatic ambiguity* - When a statement is not specific, and the sentence does not provide the information needed to clarify the statement's meaning.

Although ambiguity remains one of the major challenges nowadays, the literature shows that several methods have been developed to reduce the impact of this problem, such as Interactive Disambiguity, Minimising Ambiguity, and Weighting Ambiguity (Khurana et al., 2017). However, this document will not focus on those methods.

Another challenge for NLP is perceiving metaphors; it is difficult for computers to differentiate between the word's true meaning and an implied something that is not ordinarily associated with the terminology (Barnden, 2008).

Using comparison abbreviations, shortened forms of a written word or phrase can also cause problems in NLP since these words increase the number of words that must be analyzed (Asogwa et al., 2007). It is also important to acknowledge the impact of misspelling words, like abbreviations;

these typing errors increase the number of words and make it difficult to obtain a decent output for NLP tasks (Hu et al., 2020).
Finally, there is still the relevant fact that in textual information, it is impossible to perceive non-verbal communication such as facial expressions and physical gestures (Pathak and Thankachan, 2012). Even if this point is obvious, it is essential to note that non-verbal communication is crucial in human communication and that many prominent technology companies are trying to develop methods to introduce non-verbal communication into their platforms (Maloney et al., 2020).
Furthermore, several disciplines are needed for a complete understanding of the NLP field, such as a good understanding of computer and information science, linguistics, psychology, and mathematics (Parveen et al., 2013). More recent developments also introduced the need to understand artificial intelligence and robotics. However, this need is later compensated by the broad applicability of NLP in numerous areas such as machine translation, summarization, cross-language information retrieval, and information extraction, among many others (Weiscbedel et al., 1989).
Although there are many different applicable tasks in NLP, this document will focus on text classification. Moreover, these NLP tasks cannot be implemented without the previous treatment of the data.

2.4 Data Pre-processing

The most critical and time-consuming step is pre-processing raw data before trying any NLP task. As large amounts of data are highly susceptible to noise, missing values, and inconsistencies, this step aims to transform raw unstructured data into a cleaner and more precise form (Jiawei Han, 2014).
This step is essential as it has a significant impact on the overall quality of the patterns mined. If this step is not made correctly, it will occur the phenomena of "garbage in garbage out" (Irfan et al., 2015), where the output of the process will present non-significant results. Data pre-processing is composed of different data management techniques that aim to improve the data in different ways. Firstly, raw data comprises several missing values, noise, outliers, and inconsistencies that should be smoothed or eliminated. This procedure is called data cleaning. Then, the size of raw data might be a problem; if the volume of information is too big, it might be interesting to implement Data Reduction strategies to similar aggregate information, eliminating redundant features, or even cluster data. On the other hand, if the volume of raw data obtained in a source is too small, it might be interesting to implement Data Integration strategies such as integrating new databases or files from different sources into a coherent database (Jiawei Han, 2014).
However, in-text mining and specific techniques can be implemented at a morphological (individual words), Table 2.1, and syntactic level (logical meaning of a sentence), Table 2.2.
The real world is composed of raw textual data that contains typing errors, lacks attributes, is inaccurate and inconsistent, and often misses information. It is impossible to mine raw data and obtain good results. After pre-processing the raw data and storing it, we are interested, as previously described, in mining patterns and relationships.

Table 2.1: Pre-processing techniques at a morphological level. Adapted from Irfan et al. (2015)

Remove stop-words	Reduces the number of words, removing words such as 'the,' 'a', or 'an, that do not impact the output. Removing stop-words improves the efficiency and effectiveness of text processing.
Stemming words	Reduces a specific word to his root form. It improves the performance of text processing.
Lemmatization	Very similar to stemming. However, it considers the context and converts the word to its meaningful base form, and it is done mainly by comparing words with a database.
Tokenization	Split words in a sentence into a vector and removes punctuation.

Table 2.2: Pre-processing techniques at a syntactic level. Adapted from Irfan et al. (2015)

Part-of-speech tagging (POS tagging)	Adds the grammatical context of a single word. In other terms, it classifies the words into nouns, verbs, adjectives, or adverbs.
Parsing	Represents the sentence into a tree-like structure and examines the grammatical construction.
Keyword spotting technique	Determines the keywords from a sentence based on WordNet-Affect (a lexicon for words).
Semantic network	Represent relationships between different concepts, events, and relationships.

2.5 Data mining

Data mining is the key step of text mining, and it consists of learning and extracting information. In this step, it is imperative to have a good idea of what is the final objective of the mining for choosing the most suitable algorithms for obtaining the desired outputs (Asogwa et al., 2007). Since the primary aim of this dissertation is to classify data, the following chapter will focus on works that have been developed with the final objective of classifying text into different categories. In the literature, several approaches are described to perform the task of classifying text into different categories. However, this section will only cover those that obtained the best results and consequently have more importance in literature. Considering the various attempts, it is possible to distinguish three successful approaches: rule-based in Section 2.5.1, machine learning-based in Section 2.5.2, and hybrid systems in Section 2.5.4.

2.5.1 Text classification using Rule-Based systems

Rule-based classification is the act of classifying text by using handcrafted rules. These classifiers use systems that make decisions based on IF-ELSE rules of semantical elements to identify words or patterns in text that can be relevant to a specific category. These systems have been used for several problems in text mining (Chakravarthy et al., 2008). However, as previously noted, this section focuses on works related to text classification.

Hoch (1994), created the INFOCLAS system to classify German business letters into corresponding categories. In this work, index terms were extracted and weighted accordingly to their relevance for each category in a manually created list of specific words. Even if this work has shown to be a step-in text classifying, the author describes how only 57% of the messages had been correctly classified.
More recently, other works were interested in creating an automatic word generator that produced those lists. Li and Yamanishi (2002) generated lists according to a sequence of IF-THEN-ELSE rules. This work shows an improvement over the more traditional rule-based methods not specified by the author. Nahm and Mooney (2002) developed a clustering method where the rules were created accordingly with more specific database domains. The author describes an improvement of around 18% in document header classification. A more recent study developed by Chakravarthy et al. (2008) presents several rule-ordering algorithms that perform better than manually written rules. The author claims that these algorithms increase the accuracy of these types of classifiers by 24%.
Even if the earlier works show a shy performance, the literature has shown that the more recent rule-based classifiers can, in fact, obtain decent results in more general categories and short-text classifying tasks. However, these classifiers make their decisions based on semantic elements and give less attention to the syntactical and lexical level of text. On the other hand, with the exponential development of computational technology, the problem of a non-supported high computational task has become less of a concern, and the former field of machine learning re-emerged.

2.5.2 The concept of Machine Learning for Text Classification

Machine learning is a type of *AI* (Artificial Intelligence) where the concept of searching for representations in text and predicting outputs is done without the process of manually creating rules (Chollet, 2017). Unlike NLP, which interprets written language, machine learning tries to find patterns based on observations and experimentations. This phenomenon is called learning. These algorithms guide themselves from feedback produced in each iteration.
Machine learning has proved to be a multi-faceted field capable of accomplishing good results in several applications such as image recognition, speech recognition, and traffic predictions, among many others, using several different techniques, levels, architectures, and tools (Asogwa et al., 2007). In this section, the text will be focused on how machine learning can be used for text classification.
These algorithms have four main techniques: Supervised Learning, when the data is labeled; Unsupervised Learning, when the label is not labeled; Semi-Supervised Learning, when only part of the data is labeled, and Reinforcement Learning, which obtains feedback from a simulator, rather than labels (Chollet, 2017; Asogwa et al., 2007). As Semi-Supervised learning and Reinforcement Learning are beyond the scope of this work, the following literature review will only focus on supervised and unsupervised methods.

2.5.3 Review of Supervised machine learning methods for Text Classification

Supervised Learning is a Machine Learning method that uses datasets that have been previously labeled considering some pre-defined categories (Khurana et al., 2017; Tripathy et al., 2015). It can be seen as a synonym of classification as it learns patterns from training data and predicts a class for each input in the pre-defined labels (Parveen et al., 2013; Jiawei Han, 2014).
Text classification with supervised methods has been used for classifying text in several studies. In his work, Lewis (1998) compares several works that use a probabilistic approach with a Naïve Bayes (NB) classifier and concludes that, even if the results with this method achieve respectable effectiveness, these results still fall short compared with more complex classifiers.
Several other works in literature compare these more complex classifiers for text classification. For example, Tripathy et al. (2015), classifies reviews into different pre-defined categories using and comparing the Support Vector Machine (SVM) and the previously mentioned NB classifiers. Moreover, Bo Pang compares the NB classifier, the SVM classifier and the Maximum Entropy classifier (ME) for sentiment classification (Bo Pang and Vaithyanathan, 2002). In more recent work, Parveen et al. (2013) also compares the same classifiers as Bo Pang with a K-Nearest Neighborhood (KNN) classifier for sentiment classification. These works converge to a mutual conclusion; the SVM classifier outperforms all the other classifiers in accuracy and F1 score. On this note, several other authors have only focused on using the SVM classifier for text classification (Dave et al., 2003; Whitelaw et al., 2005).
It is essential to keep tracking the state-of-the-art Machine Learning field as it constantly evolves. A great way to see how this world is developing is to follow the highly competitive competitions on Kaggle. Nowadays, even if there has been around for several years, the Gradient Boosting classifier is one of the most used classifiers in the competition (Chollet, 2017). In his work, Alzamzami et al. (2020), compares several classifiers for sentiment classification, among them the SVM classifier, several Gradient Boosting classifiers, and Deep Learning algorithms. The studies of his work conclude that Gradient boosting classifiers overperform the SVMs and converge quicker than Deep Learning algorithms.
Summarizing, the literature shows that SVMs and Gradient Boosting classifiers are the appropriate tools for text classification using Machine Learning techniques. However, it also opens the door to a new concept: Deep Learning.

2.5.4 The concept of Deep Learning for Text Classification

The idea of deep learning is a subfield of machine learning that saw its birth in the early 1950s; however, due to a lack of computational capacity, it only rose as an effective field in the early 2000s (Chollet, 2017).
Deep learning is a mathematical structure for learning patterns from data. Chollet (2017) describes deep learning as a “multistage-distillation operations where information goes through successive filters and comes out increasingly purified”. In other words, deep learning is based on a structure composed of several successive layers that apply mathematical transformations in the data

according to the weights of each layer, trying to find the best representation of the input data. The number of layers is the depth of the model.
The learning phase is the process of finding the most accurate value for the weights in each transformation so that the network is capable of correctly classifying the data into their corresponding label (Chollet, 2017). A loss function measures how well the network is classifying the given data in each iteration. This function compares the prediction of the network with the genuine labels and designs a numeric path that the algorithm should follow so the objective of minimizing the function is accomplished.
Deep learning is a fascinating field because, as Chollet points out, it fully automates the step of manually building good layers (feature engineering). Rajpoot et al. (2021) describes in their work that the problem of classifying large amounts of data into predefined categories can be overcome by using deep learning-based approaches such as DBN (Deep Belied Network), RNN (Recurrent Neural Network), CNN (Convolution Neural Network), and Hybrid Models. Furthermore, Minaee et al. (2020) evaluates and compares several deep learning methods with machine learning methods for text classification. Among the multiple methods used, the more relevant use RNN-Based Models, CNN-Based Models, models with Attention Mechanisms, and Hybrid Models. Having tested these methods in publicly available datasets, Minaee et al. (2020) concluded that deep learning methods perform better than machine learning SVMs and Gradient Boosting methods, even if the difference was insignificant.
On the same note, Yechuri and Ramadass focused their work on deep learning RNN and Long Short-Term memory networks (LSTM) for predicting sentiment in movie reviews. This study demonstrates how LSTM models are better predictors than existing models for classifying sequential text data (Yechuri and Ramadass, 2021).
Focusing on hybrid models, S. Al-Deen's research addresses problems related to CNN and LSTM of data sparsity by combining recurrent bidirectional long short-term memory units (Bi-LSTM), CNN, and multi-head attention (DNN–MHAT) mechanisms (Al-Deen et al., 2021). The results from their research show how the method DNN-MHAT overperforms more simple deep learning techniques.
The literature has shown that multiple Deep Learning techniques could be successfully used and even outperform other Machine Learning techniques for classifying text into several pre-defined classes. It can be noted that the more interesting methods are LSTM and hybrid methods such as DNN-MHAT.

2.6 Models used in Supervised Learning

This section is dedicated to explaining the three methods highlighted for text classification with Supervised Machine Learning. On this note, the first approach was the Support Vector Machine (SVM), the second approach uses the Gradient Boosting method, and finally, the last approach considered is the development of a Deep Learning (DL) model.

It is important to underline that for the following sections the variable x will correspond to the variable *Review Text* and y to the variable *Overall.Rating* from Table 3.1 to simplify the representation of formulas.

2.6.1 Support Vector Machine (SVM)

This method is a well-known process supported by extensive mathematical theory, and it can handle both linear and nonlinear classification problems using kernel functions. SVMs are based on the simple concept of dividing the data into several classes by creating *hyperplanes*. These *hyperplanes* are mathematical functions that represent a decision boundary that separates the data by an *Optimal hyperplane*. Furthermore, they use a *Maximal margin* to minimize a tradeoff between empirical error and the complexity of the space analyzed. Figure 2.2 gives a simplified idea of what each element is.

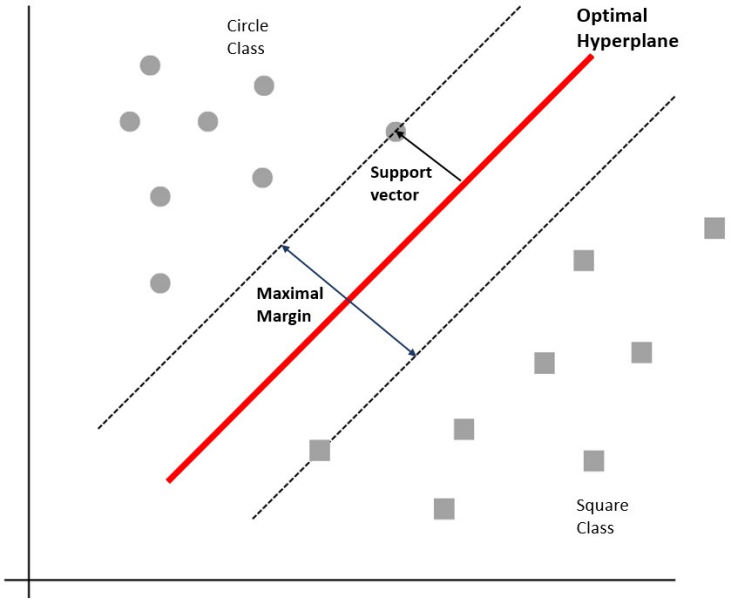


Figure 2.2: Hyperplane representation. Adapted from Pupale (2018)

SVMs, in the simplest way, do not support multi-class classification as the hyperplanes separate the space into two different classes. However, they can be adapted for multi-class classification using the same principle and breaking up the problem into multiple binary classification problems. There are various ways of processing this step; however, as the *LinearSVC* SVM will be used, this step is made by using a One-to-Rest multi-class strategy approach dividing the dataset into multiple binary classifications, training on each binary classification, and predicting the membership of the text in one of the five classes.

On the same note, this method has two main steps: the first one is to map the data into a dimensional representation, and the second is to create decision boundaries.

The process of mapping the data into a dimensional representation is done by interpreting the encoded information after the encoding step. On the other hand, creating the decision boundaries

means computing the separation of the hyperplanes with the training data. For each class, the goal is to maximize the distance and the margin between the hyperplane and the closest data point.
As previously described, the optimal hyperplane is the one that has the most significant distance to the nearest data point of any class; this distance is called the functional margin. To calculate this function, first is needed to find the points closer to the optimal hyperplane, called the support vectors, and then the distance between the two should be computed. This process requires adding an extra dimension so that the hyperplane becomes a linear function and then projecting the boundary in the original dimension using mathematical transformations provided by the kernels function.
Going deeper into the theoretical definition and presenting the mathematical formulation of the algorithm. These formulations are inspired in Sassano's work (Sassano, 2003).
The training data is represented by (*x_i*,*y*) , being *x_i* the encoded tokens and *y* the normalized overall rating associated.

$$(x_i, y), \ \ldots, (x_1, y), x_i \in \mathbb{R}, y_i \in \{-1, 1\} \tag{2.3}$$

The decision function *g(x)* for creating the hyperplane is given by Equation 2.4.

$$g(x) = sgn(f(x)) \tag{2.4}$$

Where *f(x)* is given by Equation 2.5.

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i k(x_i, x) + b \tag{2.5}$$

Respecting the constrains in Equation 2.6.

$$\forall_i : 0 \leq \alpha_i \leq C \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.6}$$

In equations 2.3, 2.4 and 2.5, k represents the kernel function responsible for the dimensional transformation, b is the threshold, α_i are the kernel weights associated and C is the miss-classification cost. The vectors x_i with weights greater than zero, $\alpha_i \geq 0$, are called the Support Vectors. Furthermore, for a linear classification case, as the method used is the *LinearSVC*, the kernel function is defined in Equation 2.7.

$$k(x, x_i) = x_i \cdot x \tag{2.7}$$

The process of training the SVM algorithm is to find the weights α_i and the threshold b that optimizes Equation 2.8.

$$max \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1} \alpha_1^u \alpha_j y_i y_j k(x_i, x_j) \tag{2.8}$$

Respecting the constrains in Equation 2.9.

$$\forall_i : 0 \leq \alpha_i \leq C, \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.9}$$

As seen in the different equations presented, some parameters must be defined. Firstly, the miss-classification cost, C, controls the tradeoff between having a smooth decision boundary and having a good classification of the data points. Secondly, the alpha parameter, α_i, defines the weight that a single training example has on the decision boundary.
The solution from the equations presented gives the optimal hyperplane. In other words, the optimal decision boundary, and the respective support vectors.

2.6.2 Gradient Boosting

This technique is known decision-tree-based ensemble learner that iterates solutions and bases decisions on prior 'weaker' classifiers. From these decisions it constructs a powerful final classifier that addresses the weak points from a collection of separate models. The model is updated using gradient descent that minimizes the loss function of the whole system until the maximum number of iterations is reached.
Before going deeply into the explanation of the method used for gradient boosting, it is essential to understand how decision trees are developed as they are the root of the gradient boosting method. A decision tree is a successive split of the data provided into different spaces to determine the best accuracy at classifying data. These structures are composed by a *root node* where all the information is in the same space. The information is then split successively according to several *decision nodes* that use a discrete function and allocate to each *leaf node* similar information. The tree is built until reaching a maximum depth defined by the user (Rokach and Maimon, 2008). Figure 2.3 presents a theoretical example of a decision tree and Figure A.1 in attachment A presents an example adapted to a text classification problem.

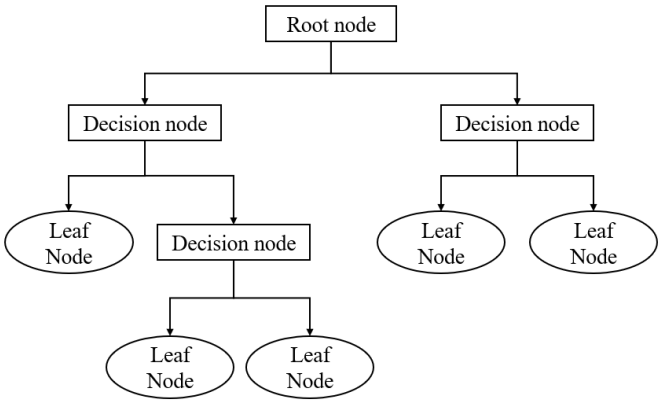


Figure 2.3: Theoretical example of a decision tree. Adapted from Rokach and Maimon (2008)

After a quick presentation of what is a decision tree, it is easy to understand how the gradient boosting algorithms work; Gradient boosting is a type of ensemble approach in which a collection of multiple decision trees called weak models are created and then combined to improve overall performance (Chen et al., 2018). In this case, the focus will be in the end-to-end open-source package tree boosting system *Extreme Gradient Boosting* (XGBoost). Furthermore, the formulation are inspired in Masui's work (Masui, 2020).
The prediction, $y\_pred_i^{(t)}$, for the t-th step is defined in Equation 2.10, being f_k the prediction from a decision tree and x_i the feature vector for the i-th data point.

$$y\_pred_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \tag{2.10}$$

To compare the result of each prediction to create a way to differentiate which decisions tree as the best result, a Loss Function, L, should be calculated; in this case of multi-classification tasks, the *mlogloss* function is used and it is defined in the Equation 2.11.

$$L = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N} y_{i,j} log(p_{i,j}) \tag{2.11}$$

In Equation 2.11, N is the number of instances, M is the number of different labels, $y_{i,j}$ is the binary variable with the expected overall rating and $p_{i,j}$ is the classification probability output given by the classifier for the i-th data point and the j-th label. Furthermore, the XGBoost objective function also includes another significant term, the regularization, Ω calculated with Equation 2.12.

$$\Omega = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{t} \omega_j^2 \tag{2.12}$$

In Equation 2.12, T is the number of leaves, ω_j^2 is the score on the j-th leaf, γ is a constant threshold of gain improvement and λ is a constant that shifts which split are taken. This regularization prevents the model from overfitting by controlling the complexity of the model for the number of leaves.
The final objective function in Equation 2.13 that should be optimized is the sum of the Equations 2.11 and 2.12.

$$Obj^{(t)} = \sum_{i=1}^{N} L\left(y_i, y\_pred_i^{(t-1)} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_t) \tag{2.13}$$

As previously documented, the first part of the objective function is the loss function measuring the difference between the prediction and the target, and the second part is the regularization that keeps simplicity. Several parameters should be defined such as the number of threads, the step size, and the regularization parameters.

2.6.3 Deep Learning

The last approach evaluated in the Supervised Learning field for text classification is developing a deep learning model. The information presented in this section is inspired by François Chollet's book, Deep Learning with Python (Chollet, 2017), and on the TensorFlow website (Abadi et al., 2015).

These models employ a layered structure of algorithms to continually analyze the data and create patterns to perform tasks such as Text Classification. Even if these end-to-end deep-learning models are challenging to understand at a deep level, their use is simple as it requires little manual work because they automate the step of feature engineering, that is, manually engineering good layers for the data.

This network architecture is composed of several compatible layers that are successively chained together to implement a process of data distillation. Layers can be imagined as a filter that accepts specifically shaped tensors to mathematically transform the tensor into a new one with refined data. As some concepts might be turbid, the structure of a deep learning algorithm is based on the structure of an artificial neural network. The only difference is in the depth of the model, as a deep learning model should contain more than three layers between the input and output layer. Figure 2.4 represents a deep learning network's quick and simplified theoretical architecture.

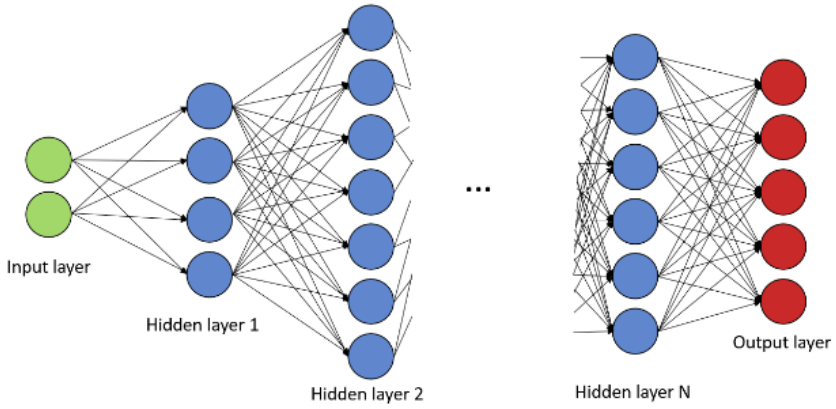


Figure 2.4: Simplified theoretical architecture of a Deep Learning network. Adapted from Afshine Amidi and Shervine Amidi (2020)

Such as the machine learning algorithms, these networks should be submitted to multiple trainings; in this case, the objective is to adjust the weights of each transformation layer to obtain a more accurate representation of the data. In these types of training loops, the weights of each transformation layer should be updated to reduce the value of the Loss Function successively. These weights are updated by computing the loss gradient, the derivative of a tensor operation, and moving the value of the weights in the opposite direction from the gradient.

It is important to emphasize that, in the model's initial, the weights in each layer are randomly assigned. Figure 2.5 gives a quick representation of the specific training loop for DL.

Going deeper into detail, as previously mentioned, finding the proper structure for the network is more an art than a science (Chollet, 2017). It is a time-consuming iterative process where the objective is to find the structure that can better predict the data.

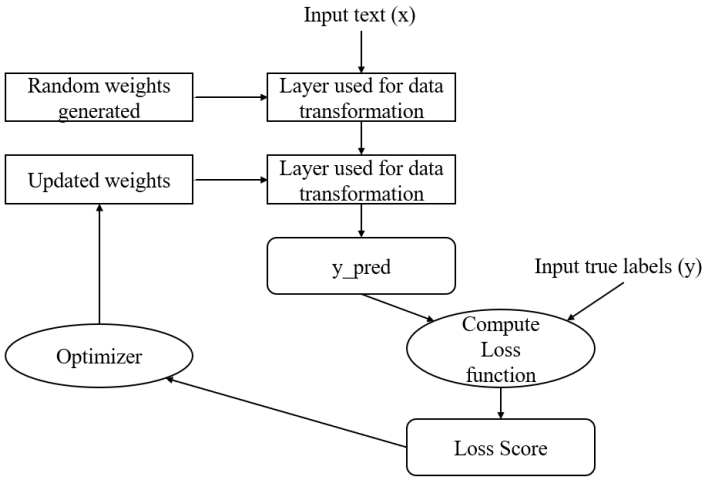


Figure 2.5: Training Loop in DL. Adapted from Chollet (2017)

The first layer used was the *Embedding Layer*; this layer maps the human language early transformed in tokens into a geometrical space. For each iteration of the training loop, this layer updates each tensor in the defined space by looking up the more appropriated values.
The second type of layer used was the *1D Convolution Layer*, responsible for learning the hierarchies of patterns and recognizing local patterns in different positions by making convolution operations. Several convolutional layers form a Convolutional Neural Network. Equation 2.14 is the function used for this transformation.

$$New\_tensor = bias\,(C_{out\ j}) + \sum_{k=0}^{C_{in}-1} Weight(C_{out\ j},k)Input(N_i,k) \tag{2.14}$$

$$k = \frac{groups}{C_{in} * kernel\_size} \tag{2.15}$$

Where, N is the batch size, C is the number of channels, the *Weight* is the learnable weights, the *groups* is the number of blocked connections between channels, the *kernel_size* is the size of the convolving kernel, and the *bias* is the learnable bias that will be changed in each iteration.
The third type is the *LeakyRelu* Layer; this layer fixes the gradient death as it does not have a zero-slope part. Equation 2.16 presents the transformation applied to the data.

$$f\,(y) = \alpha y,\ \ y < 0 \tag{2.16}$$

The fourth type of layer is called the *MaxPooling1D* layer and the *GlobalMaxPooling1D*. These layers have the objective of down sampling the number of features mapped selecting from each batch the maximum value present, consequently down sampling the number of feature-map coefficients to process. The difference is that the *MaxPooling1D* layer should be used after a *Convo-*

lution layer. Figure 2.6 gives a visual representation of the transformation made by the two layers presented.

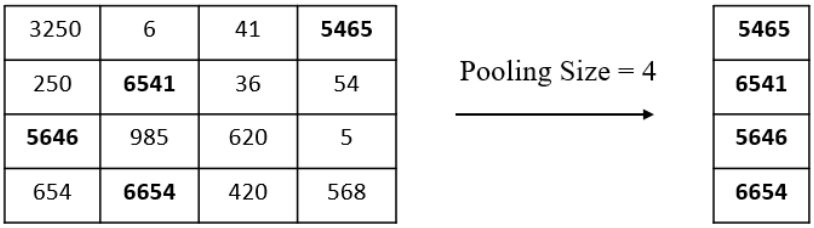


Figure 2.6: Transformation made by the *MaxPooling1D* layer. Adapted from Gite et al. (2021)

The fifth type of la used is the layer *Bidirectional LSTM* layers. The part of the *LSTM* (Long Short-Term Memory) layer is referent to a kind of recurrent neural network and aims to save the vanishment of older information in the training process. The bidirectional part is the fact that the layer can store information from the recurrent network in different ways as the input flows in two directions. The *LSTM* layer requires a more profound explanation to be understood, with this intuit Figure A.2 in Attachment A presents the *LSTM* structure for an easy example and a deeper explanation of the concept.

The sixth type of layer is the *Dense Layer* that applies a matrix-vector multiplication to change the dimension of the tensor; in other words, this layer uses mathematical transformations such as rotations, scaling, and translations in a tensor with the objective of grouping data into a specific neuron. The parameters used for each transformation are trained and updated in the training loop. As it groups information, this layer is usually used as the last layer to obtain the final results; in the case of this dissertation, the last layer will output a 5-dimensional vector. The dense layer can be represented by the right side of the Figure 2.4.

The final type of layers used for constructing this model is the *Dropout* Layers used for making the network different in each iteration, diminishing the probability of overfitting the model. These layers work by removing different some neurons during the training step. In each iteration, the selected neurons removed are different. Figure 2.7 presents a visual representation of the Dropout layer.

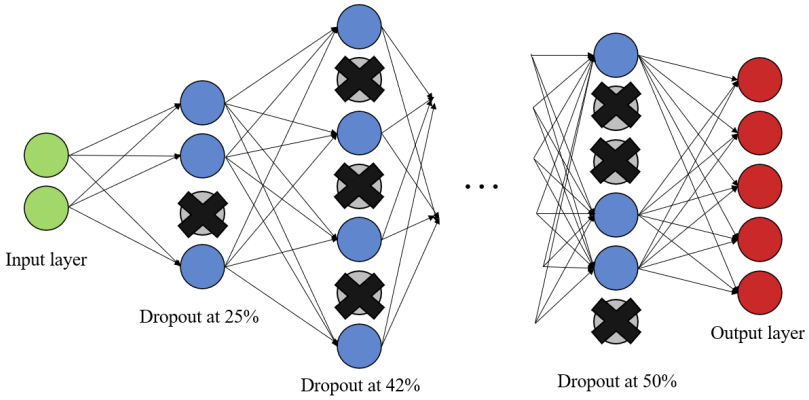


Figure 2.7: Visualization for the Dropout layer

Furthermore, each of these layers should be activated by an activation function; the function constrains the range of the input and output that the neuron can access.
The final parameter that was defined in this deep learning structure was the optimizer used; the objective of the optimizer is to find values for the parameters where the loss function attains the lower value. This parameter can make the difference between the algorithm converging to a solution or not converging. The optimizer used for this task was the *RMSprop* optimizer. Equation 2.19 present the calculation that updates the weight of each parameter.

$$v_t = \rho v_{t-1} + (1-\rho) * {g_t}^2 \tag{2.17}$$

$$\Delta\omega_t = -\frac{\eta}{\sqrt{v_t + e}} * g_t \tag{2.18}$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t \tag{2.19}$$

In Equations 2.17, 2.18 and 2.19, the parameter η corresponds to the initial learning rate, v_t is the exponential average of squares of gradients, g_t is the gradient at the time t, ρ is a hyperparameter, e is a parameter used only to make sure the formulation is not divided by zero, ω_t is the weight of each parameter at time t and ω_{t+1} corresponds to the update weight of each parameter.

2.6.4 Evaluation methods

To evaluate and compare the models, it is necessary to define functions that represent how well the algorithm is performing; in other words, the function needs to calculate how many *Overall.Rating* labels, y, are predicted accurately.
There are several metrics that can be used for multi-class classification, in this case the functions used were the Accuracy Classification Score (*accuracy_score*), the *Precision*, the *Recall* and the *F1-score*. Furthermore, a *confusion matrix* was developed for an easier calculation and comparison between algorithms.
Firstly, the *accuracy_score*, presented in Equation 2.20, computes the fraction of correct predictions in the sample. In equation 2.20, *N_samples* is the total number of samples, y_i is the real

Overall.Rating classification label for input i and y_pred_i is the classification label predicted by the algorithm.

$$accuracy\_score\,(y, y\_pred) = \frac{1}{N_{samples}} \sum_{i=1}^{N\_samples-1} 1 * (y_i = y\_pred_i) \tag{2.20}$$

Furthermore, a multi-label *confusion matrix* was develop where the number of predictions class-wise is presented. In other words, this matrix divides the different classes accordingly to the true label of the input data and then computes the number of predictions for each label. For a better visual interpretation of the *confusion matrix*, Table 2.3 presents a theoretical example of three possible label classifications.

Table 2.3: Theoretical example of a confusion matrix

y_i=1 & y_pred_i=1	y_i=1 & y_pred_i=2	y_i=1 & y_pred_i=3
y_i=2 & y_pred_i=1	y_i=2 & y_pred_i=2	y_i=2 & y_pred_i=3
y_i=3 & y_pred_i=1	y_i=3 & y_pred_i=2	y_i=3 & y_pred_i=3

The *Precision*, the *Recall* and the *F1-score* metric can easily be calculated as all the values are present in the *confusion matrix*. The *Precision* refers to the ability that the classifier has to correctly predict the class between the total number of the true labels in that same class. On the other hand, the *Recall* function measures the correct predicted labels between the total number of predictions for that class. Finally, the *F1-score*, presented in Equation 2.23, is the weighted harmonic mean of the precision and the recall and can be easily calculated using the values obtained previously.
These metrics were automatically calculated using the *macro average* method and the *weighted* method; however, we will only use the *macro average* method for comparing the results. The *Precision macro average*, Equation 2.21, is the mean of all the precision scores of different classes, on the same note, the *Recall macro average*, Equation 2.22 is the mean of all the recall scores of different classes.

$$PrecisionMacroAvg = \frac{Prec_1 + Prec_2 + Prec_3}{n} \;\; and \;\; Prec_N = \frac{TP_N}{TP_N + FP_N} \tag{2.21}$$

$$RecallMacroAvg = \frac{Recall_1 + Recall_2 + Recall_3}{n} \;\; and \;\; Recall_N = \frac{TP_N}{TP_N + FN_N} \tag{2.22}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.23}$$

In Equations 2.21, 2.22 and 2.23, n represents the number of possible classes, TP_N represents the true positives for class N, in other words, the number of predictions that the algorithm correctly predicted the label for the class, FP_N represents the false positives for class N, the number of incorrectly predicted labels that were the right label, and FN_N represents the false negatives for class N, the number of incorrectly predicted labels that being the wrong label.

Furthermore, for the DL method the metrics used for evaluation were the *accuracy_score*, previously described, and the *Sparce Categorical Cross Entropy Loss* presented in Equation 2.24. This metric calculates a score using the average difference between the actual label of the input data and the predicted probability distributions for the different classes.

$$Sparce\ Categorical\ Croos\ Entropy\ Loss = -\frac{1}{N}\sum_{i=1}^{N\_output\_size-1} y_i * log(y\_pred_i) \tag{2.24}$$

In Equation 2.24, y_i is the real *Overall.Rating* classification label for the input i, N is the number of predictions and y_pred_i is the label predicted for the input text i.

2.7 Unsupervised Machine Learning methods for Text Classification

Extracted datasets are not usually previously labeled according to the categories in which we want to classify our data. In addition, transforming an unlabeled dataset manually into a labeled dataset is a complex and expensive task. Also, the text is difficult to classify as it might depend on personal interpretation and opinion. Unsupervised classification is a critical method in these occasions as it amalgamates the information that comes closest.

Two main types of techniques are used for unsupervised text classification: Association (Ko and Seo, 2000; Bo Pang and Vaithyanathan, 2002) and Clustering (Shafiabady et al., 2016; Hayat et al., 2021).

Early works of unsupervised text classification use association rules; as the name suggests, the algorithms search for predefined rules in the dataset that can be associated with a class. Ko and Seo (2000) proposed an association rule method that classified sentences accordingly with a manually pre-defined list of keywords for each category. The author compares the defined method with a supervised learning method and concludes that the two methods obtained similar performances. Furthermore, Turney presented an automatic classification based on Part of Speech (POS), where adjectives and adverbs were used to estimate the average semantic orientation of the phrase. This algorithm obtains an average accuracy of 74% (Bo Pang and Vaithyanathan, 2002).

Other works opted for clustering techniques, Hayat et al. (2021) obtained good results classifying news headlines using Mean Shift and K-means algorithms as self-learning automatic classifiers for feeding a supervised Machine Learning algorithm. Moreover, N. Shafiabady et al. (2016) produced automatic clusters using Correlation Coefficients and Self Organizing Maps (SOM) for feeding a supervised SVM. He concluded that combining unsupervised, clustering text, and SVM supervised machine learning obtains better accuracy than manually classifying a dataset. Lee and Yang (2009) used Latent Semantic Indexing (LSI) to reduce the data dimensions for training the SOM clustering. The conclusion from this work is that the difference was not significant.

There are multiple unsupervised methods for text classification that have proven to produce interesting results. However, several are not fully automated ways of classifying text, and their results are then used for training a supervised algorithm. This is where the use of innovations like transformers can be interesting.

2.7.1 Transformers method for Unsupervised Text Classification

The introduction of Transformers is recognized as the most successful and significant development in modern NLP, overcoming several limitations related to implementing DNNs. Lee and Yang comment on how CNNs are not excellent at processing sequential data and how long sequences are challenging for RNN to process (Ma et al., 2020). On the other hand, transformers overcome these constraints by using self-attention mechanisms to model the influence of each word on another word, by being trained in massive amounts of data from different sources, and by allowing parallel training (Ma et al., 2020; Minaee et al., 2020). Even if these models seem semi-supervised as they are trained in data, they will be used as a complete unsupervised method in this dissertation.

These Transformer-based Pretrained Language Models (PLMs) can be implemented in unsupervised tasks, e.g., Zero-Shot Classification (Chen et al., 2021), and semi-supervised tasks, e.g., Few-Shot Learning (Geng et al., 2019), as a sentence encoder for textual information.

Analyzing the literature related to Transformers, it can be stated that these models have become the state-of-the-art for NLP problems and that the more used and explored models are the BERT model and the ELMo model (Chen et al., 2021; Minaee et al., 2020; Ma et al., 2021). However, in the following section, the focus will be on the Bert model.

2.7.2 BERT model

Jacob Devlin and his co-workers at Google presented in 2018 the first BERT (Bidirectional Encoder Representations from Transformers) model that was later implemented in the search engine Google. BERT is based simply on attention mechanisms; however, it overperforms several other DNNs (Vaswani et al., 2017). As Minaee et al. (2020) describes, this Transformer is trained by randomly hiding tokens in text sequences and then recovering them based on the encoding vectors obtained, called the Masked Language Modeling (MLM). To better understand the structure of the Bert model, it is recommended to read the full paper released by Google.

Over time, several derivatives of the first Bert model appeared that focused more on developing a specific task. Among others, it is interesting to highlight those set for NLP-related tasks and text classification, such as the S-Bert model, BERT-CLS, S-BERT-Glove and BART-NLI, and the RoBERTa (Chen et al., 2021; Ma et al., 2021).

2.7.3 Zero-Shot Classification with BERT

Datasets that have not been previously labeled are challenging to classify. However, PLMs can be used as encoders to classify data in Zero-Shot Classification.

Chen et al. (2021) describes how he used the pre-trained sentence-based BERT model (S-BERT) to represent tweets into embedding spaces for later classification and documented how PLMs archive state-of-the-art results. Furthermore, Ma et al. (2021) tested several Bert-based models and supported the idea that the performances of these models are remarkably promising.

Furthermore, to evaluate the different models, Ma et al. (2021) calculated the number of exact

correctly labeled texts and reported the fraction of incorrectly predicted labels (Hamming loss). Having obtained good results with Zero-Shot learning, Ma et al. (2021) describes how it plans to use the future using few-shot learning for text classification by manually labeling a sample of data.

2.7.4 Cosine Similarity

The literature shows that for the method, Zero-Shot Classification, the final values of an embedded sentence are calculated using the cosine similarity, presented in Equation 2.25 (Reimers and Gurevych, 2019).

$$\hat{c} = argmax\, cos(\phi_{sent}(sentence), \phi_{sent}(C)) \tag{2.25}$$

In Equation 2.25, *cos* is the cosine similarity, ϕ_{sent} represents the embedding model used, the *sentence* is the review that should be embedded, the C stands for the set of possible class names. Moreover, the $\hat{c}$ is the result representing the similarity of the sentence to a particular class having a number between 0 and 1; this number represents the probability of being from a specific category. In the literature, it was not documented any type of evaluation method specific for the Zero-Shot Classification method, so the methods used in Section 3.6.3 are inspired in the work of Nagesh Singh Chauhan (2021).

Chapter 3

Work Development

This chapter presents how the company deals with reviews and the datasets used. Then, it describes the different steps implemented to prepare the final datasets, previously described in Section 2.4, as well as the different libraries used and parameters defined for each model used for Supervised and Unsupervised Classification. Finally, it introduces the evaluation methods implemented for the Unsupervised classification task.

3.1 The company vis-a-vis the technology and the reviews treatment

This section has the objective of presenting, at the time of development of the dissertation, the current state of the company in terms of the number of reviews on the existing products and how Worten treats this information.

The dataset provided by the company presents 37 695 products with at least one review. This number is composed of Worten products corresponding to a total of 37 501 products and marketplace products corresponding to 194 products. Even if the number of products with at least one review from marketplace is low, the number will increase as the marketplace's introduction can be considered recent, and the number of marketplaces' sellers and products is expected to increase as described in Section 1.3.

Furthermore, it is important to understand what is the weight of products with a low number of reviews in the platform. Products with 2 or fewer reviews were counted in the dataset. The results of this analysis were impressive since 63,8% of products from Worten, and 98,9% of marketplace products contained two or fewer reviews.

The impact of a poorly rated review is enormous on the overall rating of products with a reduced number of reviews. Consequently, the product's positioning is negatively affected on the platform. For example, a poorly classified review in an item with only two reviews can quickly convert a 5-star product into a two-and-a-half-star product resulting in the disappearance of the product from the first positions of the page. Figure 3.1 presents an example of a poorly rated review.

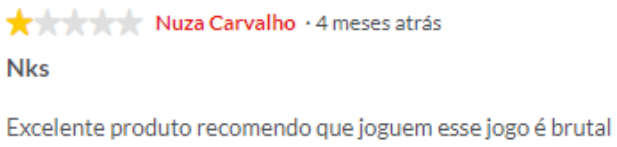


Figure 3.1: Example of a poorly rated review present in the company's platform

The review translated to English is: "Excellent product. I recommend you to play this game is brutal"; However, the customer only rated the product with one star. The number of times this happens is very difficult to quantify; however, this aspect becomes worrisome due to the high number of products with a reduced number of reviews.
Furthermore, at the moment, the company does not have any work in progress, nor does it use any outsourcing program to deal with these problems. The company only accounts with a program that detects fake reviews. This program makes automatic previsions for fake reviews, and these exact predictions can later be changed manually. However, it does not detect if the rating given by the consumer is coherent with the text.
Moreover, consumer reviews are extremely important to understand what is going on in several company fields. Currently, there is no internal or external way of making a quick analysis of the reviews, and the company does not rely on any type of program capable of classifying reviews into multiple categories depending on the topics covered.
After knowing the company's state regarding the reviews, some aspects can be improved. To compare the rating given by the consumer and the written text, it is proposed the creation of a textual rating through Supervised Learning methods using the dataset provided by the company. Furthermore, with the dataset provided by the company, the use of the Unsupervised Classification method is proposed to develop a plan to classify the reviews into different categories. Finally, it was suggested the development of a dashboard where both the reviews and the predicted results can be easily analyzed.

3.2 The Datasets

Creating a dataset is essential in Supervised ML and DL methods as they require input data to train and test their algorithms. On the other hand, Unsupervised Classification also needs a dataset for results evaluation. For this reason, it is described and commented in this section the datasets used for training and validating the different methods.

3.2.1 Data Collection

In this section, a dataset was created from the reviews extracted from Worten's database. This dataset contains multiple reviews from the *Worten.pt*, *Worten.es* and *Canarias.Worten.es* websites. These three websites share the reviews to show the maximum number reviews possible, in other words, to increase the volume of reviews, for a single product to the consumer. With the same

objective, these websites also incorporate reviews from the products manufacturer's website for the most popular products.
In addition, to know how our algorithms are performing, it was decided to compare them with other works that had the same objective of text classification. If the results are close, we can conclude that our algorithm is working properly. The Coursera reviews dataset and the Amazon musical instruments reviews are used for this comparison; these two pre-labeled datasets are well-known datasets used in different classification works.

- The Amazon's musical instruments reviews is a dataset that contains around 220 000 reviews and their respective star rating (1 to 5 stars) from May 1996 until July 2014;
- The Coursera' reviews is a dataset created from scrapping the Coursera website that contains, as the previous one, reviews and their respective star rating (1 to 5 stars); this dataset is smaller containing around 100 000 reviews.

The quantity of data in a dataset is crucial when talking about Supervised Learning, especially Deep Learning. These types of methods used for classification require massive amounts of data to train their architecture correctly due to their complexity.

3.2.2 Data Exploration

As noted in Section 3.2.1, three different datasets were used for supervised learning. However, the essential dataset to analyze is the Worten's dataset, as it is crucial to better understand the information in which the work will be developed. This dataset contained more information than what was needed for the dissertation, as it can be seen in a random sample of the raw data presented in Figure B.1 (c.f. Attachment B).
Looking at Figure B.1 in Attachment B, the dataset extracted was composed of several variables such as *Id_Review*, *Review Submission Date*, *Review Display Locale*, *Campaign ID*, *Ratings-only (Y/N)*, *Overall Rating*, *Review Title*, *Review Text*, *Product ID*, among others. However, for Text Classification, the only variables that need to incorporate the new dataset are the *Id_Review*, the *Review Text*, and the *Overall.Rating*. Table 3.1 presents a random sample of the new dataset created.

Table 3.1: Random sample of 5 reviews in the new dataset created

Id_Review	Review Text	Overall.Rating
1116062491	Já vi este artigo mais de 10 vezes a dizer que ...	1
1116060659	Funciona muito bem, muito silencioso, aliado ...	5
1116059300	Lavagem Perfeita	5
1030399886	Produto muito bom e a bom preço de excelente ...	4
1030401879	Boa relação qualidade/preço e com um ótimo ...	5

In this new dataset, the variable *Id_Review* represents the identification number of the reviews; the *Review Text* variable contains the text written by the reviewer and the *Overall.Rating* represents

the current overall rating of the review on the website.
As one of the objectives of the work is to create a new textual rating through ML and DL methods, it is essential to understand the dataset distribution for each possible *Overall.Rating*. Figure 3.2 shows the current distribution of the dataset.

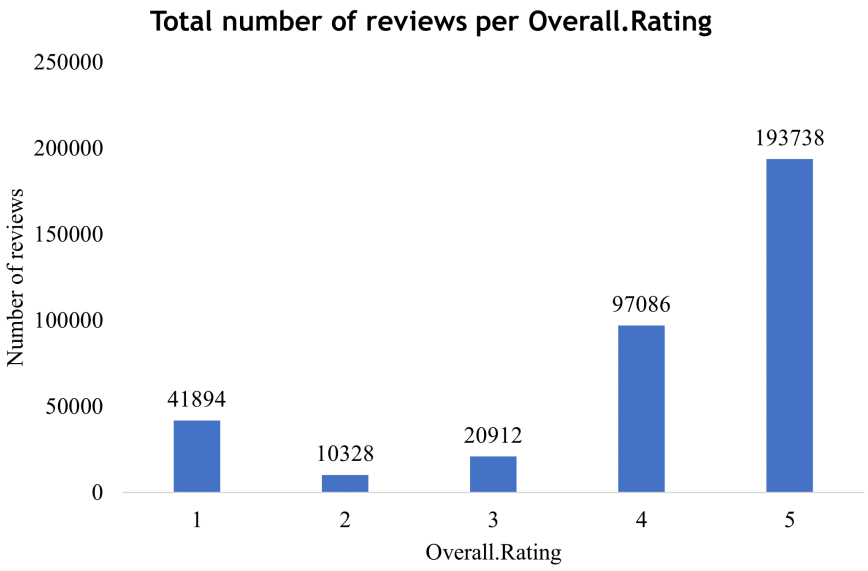


Figure 3.2: Distribution of the number of reviews per Overall.Rating

Figure 3.2 demonstrates that the number of reviews with an *Overall.Rating* of 5-stars is far superior than the others. This distribution must be considered when creating datasets for training.
Having seen the distribution of the variable *Overall.Rating*, it is now important to understand the *Text Review* variable. This variable is composed of approximately 223 000 different words and has an average of 11 words per review. Moreover, it is interesting to understand which tokens (words) appear more frequently in the dataset. For that, a term matrix was created, removing the English stop words. These stop words are presented in Table C.2 in Attachment C . Moreover, with the help of the *wordcloud* library, a graphic was also developed, Figure 3.3. This *wordcloud* visual representation gives more importance to the more frequent terms by presenting them in a bigger size than less frequent words.

Figure 3.3: Wordcloud visualization of the most frequent tokens in the dataset

The graphic obtained shows that more common words are related to describing products and services, such as *product*, *work*, *quality*, *purchase*, and *functionality*, among others. Moreover, some words related to feelings can also be perceived, such as *good*, *satisfy*, *excellent*, *recommend*, *happy*, and *great*. In general, the most common words represent a positive sentiment from the consumer; for a more profound interpretation and quicker visualization of the more frequent words in the dataset, Figure B.2 in Attachment B is presented.

Furthermore, as previously documented, Worten's dataset contains reviews from multiple sources. This topic is relevant as these sources cause the dataset to have more than one language. The detect function from the *langdetect* library was used to detect the different languages in the dataset. The results might not be a perfect detection; however, 33 different languages were detected. The most frequent languages are Portuguese, Spanish, Catalan, English, and French, as it was foreseen. Figure 3.4 presents the distribution for the reviews present in Worten's dataset.

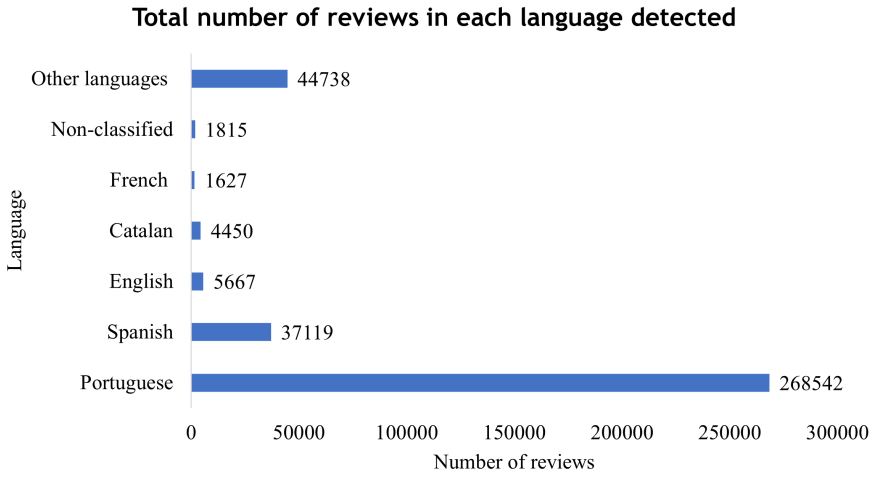


Figure 3.4: Total number of reviews for each detected language

As the *langdetect* library detected a total number of 33 different languages in the dataset, the decision to transform all the reviews into a single language was taken and described in Section 3.4. After the construction of the final dataset; it was obtained a dataset with a total of approximately 364 000 reviews.

3.3 Dataset Preparation

This section aims to explain to the reader how the problem of unbalanced data was resolved using the *Stratified Random Sampling* method and how the final distribution of each dataset was obtained.

3.3.1 Dataset preparation for Supervised Learning

As it was described in Section 3.2.2, the available datasets contain vast amounts of information and are in their nature unbalanced in terms of the number of reviews for each *Overall.Rating* scores.
A sample that captures the main characteristic of the dataset should be selected to reasonably use the information extracted for Supervised Learning. This step was made using a *Stratified Random Sampling* method where a new strata dataset was created with random samples for each dataset (Adam Hayes, 2021). As previously mentioned, these strata datasets should represent the real distribution of the principal dataset where the information was extracted. In this case, as the objective is to predict a textual rating for the reviews, the strata datasets should have similar distribution for each star classification.
In the same note, it is difficult to quantify how much input data is needed to train, test, and validate the different Supervised Learning methods, as it is impossible to find a specific number, especially for the DL method. However, to counter this difficulty and to have datasets with similar data the most common rating is limited to 50 000 reviews for the Worten and Amazon datasets as they are the most extensive datasets. On the other hand, the limit defined for the Coursera dataset was 40

000 reviews. Then, the objective was to understand the percentage of each rating in the original distribution of each dataset and apply it to the limited dataset. Finally, this percentage was rounded off to a close number depending on the needs and the more accurate representation of the reality of the dataset to prevent the different methods used from ignoring ratings with little data. Table 3.2, Table 3.3 and Table 3.4 present the total number of reviews selected using the maximum limit for each new dataset and the percentage applied after the data pre-processing step described in Section 3.4.

Table 3.2: New distribution for the Worten dataset

Overall.Rating	**Nº of reviews**	**Percentage(%)**	**New Value**	**Percentage applied (%)**
1	41894	11,51%	5000	10%
2	10328	2,84%	2500	5%
3	20912	5,74%	2500	5%
4	97086	26,67%	15000	30%
5	193738	53,23%	50000	50%
Total	363958	100%	75000	100%

Table 3.3: New distribution for the Amazon dataset

Overall.Rating	**Nº of reviews**	**Percentage(%)**	**New Value**	**Percentage applied (%)**
1	22875	10,31%	5000	10%
2	12798	5,77%	2500	5%
3	20296	10,93%	7500	15%
4	31751	14,31%	10000	20%
5	134113	55,43%	50000	50%
Total	221833	100%	75000	100%

Table 3.4: New distribution for the Coursera dataset

Overall.Rating	**Nº of reviews**	**Percentage(%)**	**New Value**	**Percentage applied (%)**
1	2404	3,57%	2000	5%
2	2193	3,26%	2000	5%
3	4941	7,34%	4000	10%
4	17767	26,4%	12000	30%
5	40000	59,43%	40000	50%
Total	67305	100%	60000	100%

Having defined the strata datasets, these needed to be divided into a training, a testing and a validation set. Table C.1 in Attachment C presents the new distribution where the training set accounts for 80% of the total data for each strata dataset, the testing and the validation set share the remaining 20% equally.

3.3.2 Dataset preparation for Unsupervised Classification evaluation

Unsupervised Classification will only be developed in the Worten Dataset as there is no need to compare the results obtained in known datasets. However, in this case, comparing the algorithm's predictions with manually pre-labeled reviews is required. This dataset for evaluation is represented by a small sample with a dimension of 50 reviews for each category. The classes in which the reviews were previously classified will be later explained in Section 3.6.1. Table 3.5 presents an example of each category in the dataset used to evaluate the performance of the Zero-Shot classifier.

Table 3.5: Sample of the dataset used for evaluating the Unsupervised Classification method

Review Text	**Category**
Deceiving stock	Stock
Really recommend the product, it is difficult to buy an equal ...	Product
Good price compare with other websites	Price
According to the description of the website	Description
Bad service provided by the marketplace seller	Marketplace
The delivery took weeks, the product must come from China	Delivery

3.4 Dataset Pre-processing

Humans communicate through language that can take the shape of spoken or written words. When spoken, some bad habits are often used for writing, and even if humans can perceive these bad habits, machines cannot understand them as they cannot comprehend text in its essence; machines only work with numbers. It is therefore required to transform this textual data into numbers in a process named encoding, described in Section 3.5.1.
However, before the encoding step, it is essential to pre-process the data as it reduces the number of tokens needed to be generated in the encoding phase and prevents the phenomenon early described of "garbage in garbage out" caused by unwanted or unimportant text that might difficult the understanding of the methods used for classification (Irfan et al., 2015). This section aims to describe to the reader the different pre-processing steps used.
For manipulating the data, the dataset was transformed into a data frame using the *pandas*' package for the Python programming language.

Drop lines that are N.A

The extracted datasets are composed of raw data known for carrying different types of errors, one of them being missing information that creates blank lines containing no information. Using the dataframe generated by using *pandas*, these lines containing no information were quickly removed so that the pre-processing process could run more efficiently. For cases in which no information is available, no textual rating or category will be assigned to the review.

Translation

As the datasets used derive from platforms open to different types of users, at first glance, the datasets contained reviews written in other languages. In the case of the Worten dataset, as previously noticed in Section 3.2.2, the presence of different languages in the datasets can be explained by the fact that Worten crosses reviews from the Portuguese and Spanish websites and some coming directly from the suppliers. In the case of the dataset from Amazon and the Coursera dataset, it can be explained by the platforms' worldwide presence.

To combat this difference in languages, it was decided to translate every review into a common language, English, as it is the language where the translation machines are more accurate. To develop this second step, the *Googletrans* python package was implemented, automatically detecting the review language and translating it to English. This package implements the Google Translate Application Programming Interface (API) and is available for unrestricted use.

Eliminate punctuation, numbers, and unknown symbols

Proper punctuation is crucial and decisive in giving the reader a better comprehension when constructing a phrase. Also, reviewers often use numbers and some symbols, such as emojis, to describe a weight or emotion associated with the reviews. However, in the case of machines, it can negatively affect the result of any NLP task as it increases the number of aspects to analyze, and they are challenging to process.

To get around this obstacle, the punctuation, the numbers, and the unknown symbols were removed using the *string.punctuation* and *string.digits* pre-initialized constants and removing every character present in one of these constants.

Convert every word into lowercases

Sentences are composed of uppercase and lowercase characters; however, machines perceive the same word written with different cases as a different one. To transform all the words of each review into lowercase, the *lower* method was used, which returns a string where all the characters are lower case.

Remove Stop Words

In human communication stop words occur in abundance. These types of words are considered low-level information and should be removed to allow the focus on crucial and more important information. Before removing these words, the sentences should be divided into single words called tokens with a *tokenizer*.

For this step, it was used the NLTK (Natural Language Toolkit) python package. This package provides easy-to-use corpora and lexical libraries. In this specific case, the *NLTK tokenizer* was used to transform the data into tokens. Later, these tokens are compared with the existing words in the stop word list of the *NLTK stop words* library, C.2 in Attachment C, and removed.

POS-tagging

The next step in pre-processing is to tag the different words in a sentence to make it easier to utilize language criteria by recognizing which portion of speech each observation belongs to. In other words, a part-of-speech (POS) tag is a specific label assigned to each token in a sentence to

denote if the word is an adjective, an adverb, or a verb. For this work, the default *NLTK' Averaged Perceptron Tagger* was used assigning the letter 'J' to adjectives, the letter 'V' to verbs, and the letter 'R' to adverbs.

Lemmatization

Finally, the word inflected form must be reduced to their root form. This step is essential because it allows more information to be discovered as several words are reduced to their root form. This process can be made by using *lemmatization* or *stemming*; however, in this case, the *lemmatization* method was used as it is a process of determining the word's lemma that depends on its context and usage within the sentence. Also, it compares each token with a word according to a dictionary. The *NLTK wordnet lemmatizer* was used for the lemmatization of words. This NLTK's built-in function compares the tokens with the existing words in the stop word list of the *NLTK wordnet lemmatizer* and replaces the words with their root forms.

Drop lines that are N.A

After all these steps, with the removal of the stop words, punctuation, unknown symbols, and numbers, some reviews might not contain any piece of information, so they should be removed.
These steps were followed by the order in which they are presented. Figure 3.5 facilitates the visualization and understanding of the different pre-processing steps using a random review from the Worten data as an example.

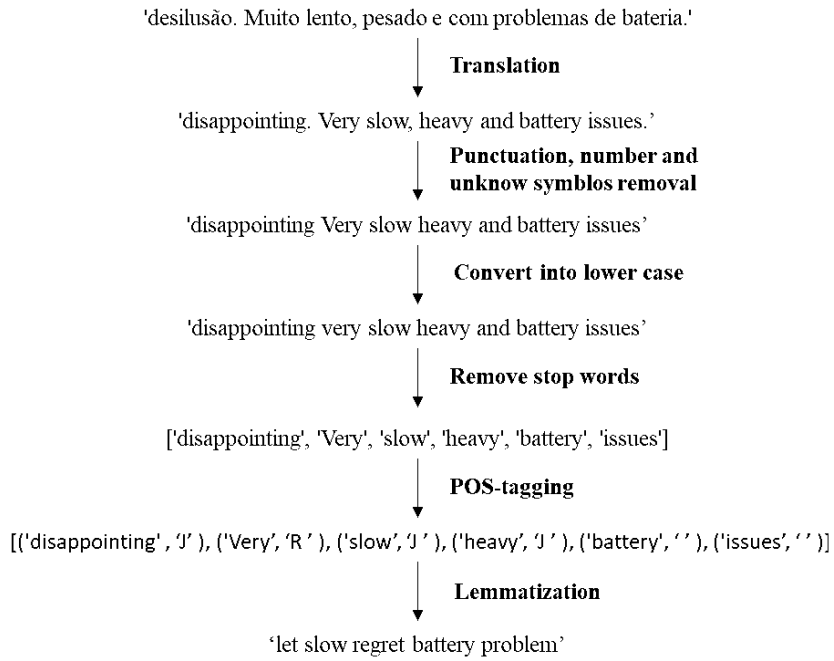


Figure 3.5: Steps used in pre-processing for an example of Worten's dataset

After the several pre-processing steps, the final Worten dataset is composed by a total of 363 958 reviews from the 386 000 available.

3.5 Description of the models used for Supervised Learning

This section aims to give the reader a deeper explanation on how the different Supervised Learning models were used to create the new textual rating. Firstly, how the information is encoded is described. Then, the different libraries used, parameters defined and the evaluation methods implemented are presented. The objective of evaluating different methods is to select which model should be used to make the *Textual Rating* predictions. These methods were used in all the previously datasets mentioned in Section 3.2.1; however, it will be chosen the best algorithm based on the Worten reviews dataset's results.
In a general way, the different methods used for Supervised Learning are based on the same iterative steps for training and obtaining results; However, the specific procedures for each method are described in Section 2.6. Furthermore, it is important to underline that in the following sections the variable x will correspond to the variable *Review Text* and y to the variable *Overall.Rating* from Table 3.1 to simplify the representation of formulas.
Summarizing, firstly, a random batch of the training set called sample of x and the correspondent y should be selected. Then, each algorithm should train their architectures with the training data and obtain predictions for the new textual rating, *y_pred*. Afterwards, the mismatch between y and *y_pred* in the selected batch should be computed using different evaluation metrics explained in Section 2.6.4. Finally, the algorithm must adjust the decisions taken to minimize the value obtained in the mismatch computed.
Yet, before implemented these steps, the textual information must be converted into numbers with an encoder, because machines cannot work with textual features.

3.5.1 Encoders

This fundamental step, called *vectorization*, consists of creating a numerical vector representing the textual information's content. Multiple types of encoders can do this step and choosing the right one is essential as it significantly impacts the result of the classifier.
For this work, the type of encoder used is the *tfidf* (term frequency-inverse document frequency) present in the *Sklearn* library. This encoder gives a statistical measure that evaluates a word's relevance to a document in a collection of sentences, in this case, in a collection of reviews. This method is revolutionary in the NLP field since it can give a better interpretation of the sentence helping to sort data into categories and extract keywords from the text as words present in different sentences will have a similar vector (Stecanella, 2019).
This encoder is based on two simple steps: Counting how many times a word t appears in a sentence d, the *Term-Frequency* ($tf(t,d)$) calculated using Equation 3.1, and calculating the *Inverse Document Frequency* ($idf(t,D)$) of the word t in the collection of reviews D, in Equation 3.2.
In Equations 3.1, 3.2 and 3.3, t is a specific word, d is the sentence where is present the word, D is the collection of reviews, in other words, the dataset where the sentences are presented, and N

is the total number of tokens in the dataset.

$$tf(t,d) = log(1 + freq(t,d)) \tag{3.1}$$

$$idf(t,D) = log\left(\frac{N}{count(d \in D : t \in d)}\right) \tag{3.2}$$

The combination of the two steps by multiplying the two vectors obtained gives the tfidf value for each token, presented in Equation 3.3.

$$tfidf(t,d,D) = tf(t,d).idf(t,D) \tag{3.3}$$

The results will be between 0 and 1. The closer the word is to 1, the rarer the word is in the dataset. The closer the word is to 0, the more frequent the word is. The objective of using this encoder is to scale down the less informative words by providing less impact to words that appear multiple times in the sentences. In other words, a word that occurs numerous times in several different sentences will have less impact than a word that appears multiple times in the same sentence but appears fewer times in the collection of reviews has this word is more specific to the sentence (Edda and Jörg, 2002).

3.5.2 Models

The literature review presents in Section 2.6 the full description of the Supervised Learning methods used. This section has the objective of describing the different libraries and parameters defined for each method tested.

Support Vector Machine

The *LinearSVC* SVM classifier was tested for multi-class classification. This method uses the One-to-Rest strategy. As it can be seen in the different equations presented in Section 2.6.1, some parameters must be defined before using the method. These parameters are presented Table 3.6.
Firstly, for the miss-classification cost, C, that controls the tradeoff between having a smooth decision boundary and having a good classification of the data points, it was assigned the value of 1, as if the value is increased the SVM classifier will not be able to generalize the data as well. Secondly, for the alpha parameter, α_i, that defines the weight that a single training example has on the decision boundary, it was defined the setting 'auto' that attributes the weight of 1 divided by the number of features. The higher the value of alpha is the more influence the closer points will have on the boundary, on the other hand, a lower number will produce a more linear curve as the points that are far away will have more weight. Moreover, the number of total features was limited to 10 thousand tokens.

Table 3.6: Description of the parameters used in the SVM classifier. Adapted from Pedregosa et al. (2011)

Parameter	Function	Value
penalty	Specifies the norm used in the penalization	l2
loss	Specifies the loss function	squared_hinge
dual	Select the algorithm to either solve the dual or primal optimization problem	True
tol	Tolerance for stopping criteria	1e-4
C	Regularization parameter	1.0
multi_class	Determines the multi-class strategy	ovr
fit_intercept	Whether to calculate the intercept for this model	True
intercept_scaling	Lessen the effect of regularization on synthetic feature weight	1
class_weight	Give a weight to each class	None
verbose	Enable verbose output	0
random_state	Controls the pseudo random number generation for shuffling the data for the dual coordinate descent	None
max_iter	The maximum number of iterations to be run	1000
kernel	Kernel function	Linear

Gradient Boosting

For testing the Gradient Boosting method, it was tested the end-to-end open-source Python package tree boosting system *Extreme Gradient Boosting* (XGBoost). Table 3.7 presents the parameters used for the *XGBoost* classifier.

In this case, the default settings were used. Furthermore, the number of total features was limited by 10 thousand tokens.

Table 3.7: Description of the parameters used in the XGBoost classifier. Adapted from Chen, Tianqi and Guestrin (2016)

Parameter	**Function**	**Value**
verbosity	Verbosity of printing messages	0
booster	Which booster to use	gbtree
validate_parameters	XGBoost will perform validation of input parameters to check whether a parameter is used or not	False
nthread	Number of parallel threads used to run XGBoost	Maximum available
dis_def_eval_metric	Flag to disable default metric.	True
num_feature	Feature dimension used in boosting	auto
eta	Step size shrinkage used in update to prevents overfitting	0.3
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree	0
max_depth	Maximum depth of a tree	5
min_child_weight	Minimum sum of instance weight (hessian) needed in a child	1
max_delta_step	Maximum delta step we allow each leaf output to be	0
sampling_method	Method to use to sample the training instances	uniform
colsample_bytree	Subsample ratio of columns when constructing each tree	1
lambda	L2 regularization term on weights	1
alpha	L1 regularization term on weights	0
tree_method	The tree construction algorithm used in XGBoost	auto
scale_pos_weight	Control the balance of positive and negative weights, useful for unbalanced classes	1
grow_policy	Controls a way new nodes are added to the tree	depthwise
max_leaves	Maximum number of nodes to be added	0
Objective	Objective Function	Multi:softmax
N_estimators	Number of parallel trees constructed during each iteration	1000

Deep Learning

For the construction of the DL model, it was used the *TensorFlow* framework for Python developed by a Google Brain Team that simplifies the defining and training of the deep-learning model. Also, for defining the different actions in the algorithm it was used the model-level library *Keras* that provides high-level blocks capable of doing low-level tasks such as operations and differentiations.

Having presented the essentials of deep learning in Section 2.6.3, we can now proceed to the explanation of the structure developed in this dissertation. In this case, the encoder was developed by the function the *Keras* function *TextVectorization* and was limited to a maximum vocabulary of 200 thousand words, and the reviews present in the dataset were limited to 100 words.

In this training loop, the *Loss Function* used was the *Sparce Categorical Cross Entropy*. This *Loss Function* should be used when the model is used for multi-class classification, and the encoder does not transform the text into binary values.

Furthermore, each of these layers should be activated by an activation function; the function constrains the range of the input and output that the neuron can access. This is an important step. The *softmax* activation function was used in the different layers. Figure D.1 in Attachment D presents the graphical representation of the function.

The structure of the DL model has iteratively changed over several attempts to obtain the best results possible. However, this process is beyond the scope of the dissertation. Figure 3.6 presents the final structure of the DL model.

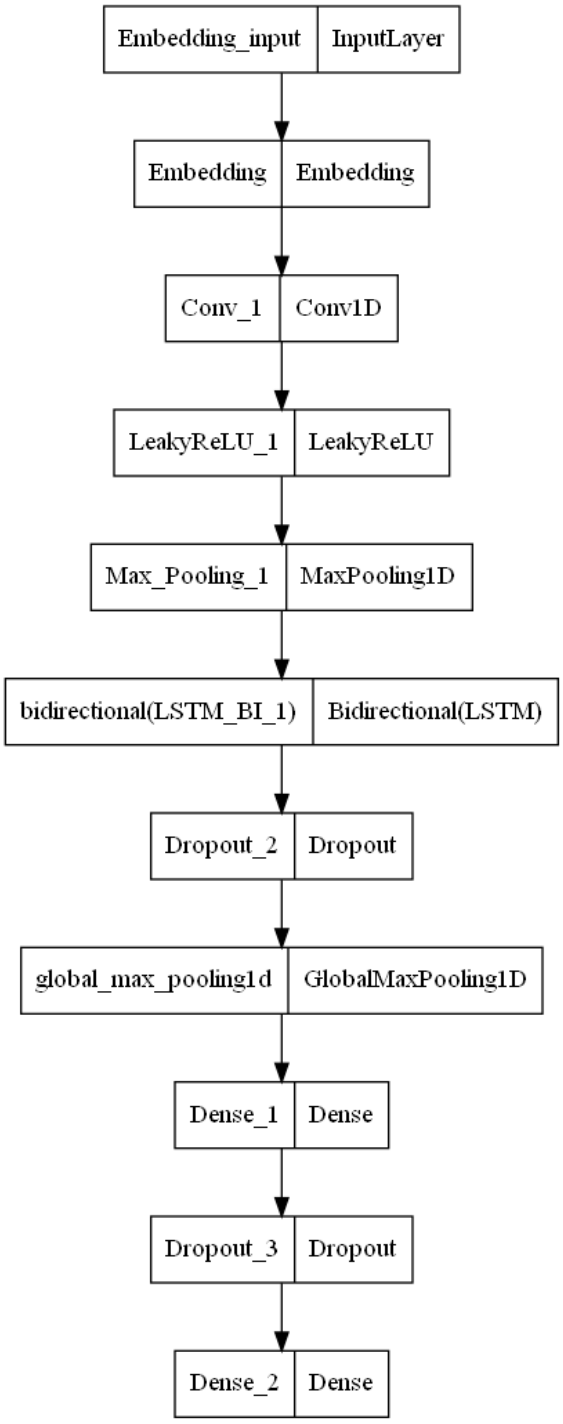


Figure 3.6: Final structure used for the DL model

Finally, Table D.1 in Attachment D presents the parameter defined for each layer of the final model.

3.6 Description of the model used for Unsupervised Classification

This section describes how the Unsupervised Classification method, Zero-Shot Classification, was used to categorize Worten reviews into different classes: *product*, *delivery*, *stock*, *price*, *description*, and *marketplace*. This classification aims to facilitate the analysis of different positive and negative aspects of each category described.
Furthermore, this section describes how the Transformer-based PLMs can be used to classify the Worten reviews. The method used was the Zero-Shot method. This method was developed using the *Pytorch* library. Here the PLMs models can be used as embedders to display the information on the same latent space and later calculate the similarity between a never seen sentence and the class name without specifying the specific class in a training step (Davison, 2020).

3.6.1 Embedding

The type of zero-shot learning used required the introduction of the classes' names; The expected results from this classification are simple. The different reviews that refer to the product's technical or more basic aspects should be classified with the label *product*. The reviews related with stock problems should be classified with the label *stock*. Furthermore, the reviews containing elements about the delivery service should be classified as *delivery*, and reviews that talk about how the price of the products is related to other websites should be classified as *price*. Moreover, the reviews about the description or the images used on the website for each product should be classified as *description*. Finally, the reviews related to the marketplace's sellers should be classified as *marketplace*. However, classifying it as a single review can be difficult because a single review can contain elements from several categories.
Many state-of-the-art classification problems use the PLM BERT as embeddings. Bert models are trained using a masked language technique where 15% of the inputted tokens are masked randomly using a mask token, [MASK], and then the network must be able to identify which word was hidden. This identification is made by choosing the word in the BERT vocabulary that maximizes a *softmax* function. This procedure enables the model to have awareness of both forward and backward context (Malmberg, 2021).
In this case, as it is a classification problem, it was searched better PLMs options than the standard BERT encoder. The option chosen was to use the Sentence-BERT (SBERT) PLM; this technique is a fine-tuning using *siamese architectures* of the BERT sequence representation that delivers richer semantics as instead of the result of the embedding being for each token, the SBERT output is a simple embedding for the entire sentence (Malmberg, 2021).
Furthermore, the default model of S-BERT uses *mean pooling* in the structure to produce the final embedding of the phrase (Reimers and Gurevych, 2019). *Mean pooling* means averaging the sentence's token embeddings. Figure 3.7 presents the structure of the S-BERT embedding. A specific S-BERT PLM was used in this method, the *cross-encoder/nli-distilroberta-base* in the *transformer*'s library. This model is available in the community and data science platform, *Hugging Face*, and it was trained to improve the classification task (Nils Reimers, 2021).

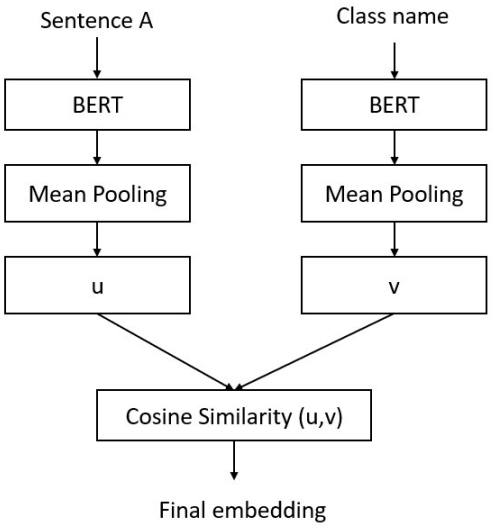


Figure 3.7: S-BERT embedding structure. Adapted from Malmberg (2021)

3.6.2 Visualization

Moreover, for representing the embedding space, it was used t-SNE visualization method with the *TSNE* tool from the *sklearn* library. This tool provides the visualization of high-dimensional data projected in a recognizable 2D or 3D space. Furthermore, for this dissertation, the sentence embedding provided by the PLM S-BERT was transformed into a 2D tensor that can be used for a 2D spatial visualization. The static visualization was developed using the *matplotlib* python library.

3.6.3 Evaluation method

Finally, to evaluate how the Zero-Shot classifier is performing, it was necessary to compare the class with closest similarity predicted by the classifier with the true label in the dataset created in Section 3.3.2. Equation 3.4 shows how it was calculated the accuracy metric for each class and then averaged it across all other classes (Nagesh Singh Chauhan, 2021).

$$\alpha_K = \frac{1}{N}\sum_{c=1}^{N}\frac{number\ correct\ predictions\ in\ class\ c}{number\ of\ samples\ in\ class\ c} \tag{3.4}$$

In Equation 3.4, K represents a set of different classes, in the case of this dissertation, K refers to the set of six different categories defined. α_K is the accuracy for the set K, N is the total number of classes and C is the specific class for which the accuracy is being calculated.
Furthermore, in an attempt to decrease the classification error a second category was added. This second category is only allocated when the cosine similarity of the first assigned category is less than 0.4 and the similarity of the second category is greater than 0.20. Equation 3.5 represents the

second accuracy calculated.

$$\beta_K = \frac{1}{N}\sum_{c=1}^{N}\frac{number\ of\ correct\ predictions\ in\ category\ c_1 and\ c_2}{Total\ number\ of\ samples\ in\ category\ c_1\ and\ c_2} \tag{3.5}$$

In Equation 3.5, c_1 and c_2 is the first category and second class assigned, and β_K is the accuracy for the set K. For this equation, reviews with second-level predictions are considered correct if: i) the first category level is correctly classified; or ii) the first level category is wrongly classified, but the second-level category is correctly classified.

Chapter 4

Results and Discussion

This section discusses the results from the methods used for Supervised and Unsupervised Classification.

4.1 Development of the Textual Rating

This subsection presents the results using the different Supervised Learning methods to create a new textual rating. The results for each model will be compared and discussed and a discussion about which method to use to create the textual rating will be drawn.

4.1.1 Results

The first objective of this dissertation is to develop a new textual rating based on the reviews' text to compare it with the actual rating of different products. To accomplish this objective, comparing and choosing the best classifier is mandatory. As such, Table 4.1 allows to compare the different metrics previously defined in Section 2.6.4.

Table 4.1: Results obtained for the Supervised Learning methods

	Method	Accuracy	Precision	Recall	F1-score
Coursera	*SVM*	72,33%	0,5	0,38	0,41
	XGBoost	72,46%	0,49	0,4	0,43
	Deep Leaning	70,92%	0,24	0,31	0,27
Amazon	*SVM*	69,83%	0,45	0,32	0,32
	XGBoost	70,44%	0,45	0,35	0,37
	Deep Leaning	68,25%	0,23	0,31	0,25
Worten	*SVM*	70,15%	0,49	0,35	0,33
	XGBoost	70,34%	0,47	0,37	0,38
	Deep Leaning	69,81%	0,24	0,32	0,27

The results presented in Table 4.1 demonstrate that the SVM and de XGBoost methods proved to be the best classifiers for the job, obtaining an average accuracy in the three datasets of 70,77% and 71,08%. On the other hand, the Deep Leaning algorithm underperformed, having an average

accuracy of 69,81%. These results for the three methods seem promising, because if the classification was done randomly, the accuracy should be around the 20% mark.
Regarding the F1-Score, the XGBoost method obtained the higher score with an average of 0,39, followed by the SVM method with an average score of 0,35. Once again, the Deep learning algorithm underperformed with an average score of 0,26.
The results are meaningless if there is no validation from existing works using the same datasets. On this note, the results obtained were compared with the work developed by Sebastian Poliak (2020) presented in Table 4.2.

Table 4.2: Comparison between the results in Sebastian Poliak (2020) and the results obtained in this dissertation

	Sebastian Results	**SVM**	**XGBoost**	**Deep Learning**
Coursera	78,73%	72,33%	72,46%	70,92%
Amazon	67,92%	69,83%	70,44%	68,25%

Several points should be considered before analyzing the results from Table 4.2. Firstly, the distribution for the training, test, and validation sets used in Sebastian's work is the same as the one used in this dissertation. However, the reviews were not pre-processed, and the entire dataset was used, unlike in this dissertation. Moreover, the method used by Sebastian Poliak (2020) for classification was a neural network composed of a bidirectional LSTM layer, a dense layer with the ReLU activation function, and an output layer using the Softmax activation function. Finally, the network was trained using the cross-entropy loss function (Sebastian Poliak, 2020).
Moreover, focusing on the Deep Learning method, the results are far from what was expected. This method underperformed in all the different datasets tested. Table 4.3 presents a closer look into the results obtained with the DL method.

Table 4.3: Deep Learning results

DL	**Training Accuracy**	**Training Loss**	**Validation Accuracy**	**Validation Loss**
Coursera	70,73%	0,8071	70,92%	0,7945
Amazon	67,71%	0,8977	68,25%	0,8991
Worten	69,76%	0,8314	69,81%	0,8343

Furthermore, the number of epochs trained to avoid overfitting the training data and the graphical representation comparing the training accuracy and the validation accuracy for the training loop is presented in Figure E.1 and Figure E.2 in Attachment E.
The results from a DL method are deeply influenced by the different structures used; however, different structured layers were tested during the development of this dissertation, and the structure presented in Figure 3.6 was the one that obtained the best results. Furthermore, the reason why the DL method underperformed is detected when analyzing the *confusion matrix* obtained. Even if the *softmax* activation was used for the last dense layer, the model only presented predictions for the 1-star and 5-star categories making this model unusable as it would only rate reviews as very

bad or very good. Table E.1 in Attachment E shows the confusion matrix obtained for the Worten dataset.

4.1.2 Discussion

The first observation that can be made is that the results obtained with the Supervised Learning method presented in Table 4.2 can be considered similar to the results obtained in the work of Sebastian Poliak (2020). The difference in the results obtained can be caused by the classification method used, the different number of reviews used for training, and in the different hyperparameters used in each method.
Furthermore, the results were satisfying, bearing in mind that textual information, especially information based on consumer opinions, as is the case of textual reviews, is extremely challenging to classify. According to the results obtained, the best method for text classification is the Gradient Boost method, *XGBoost*. As so, this method was used to make predictions for every textual review in the Worten dataset.
The absolute values for the difference between the current website rating and the new textual rating for each review was measured and presented in Figure 4.1.

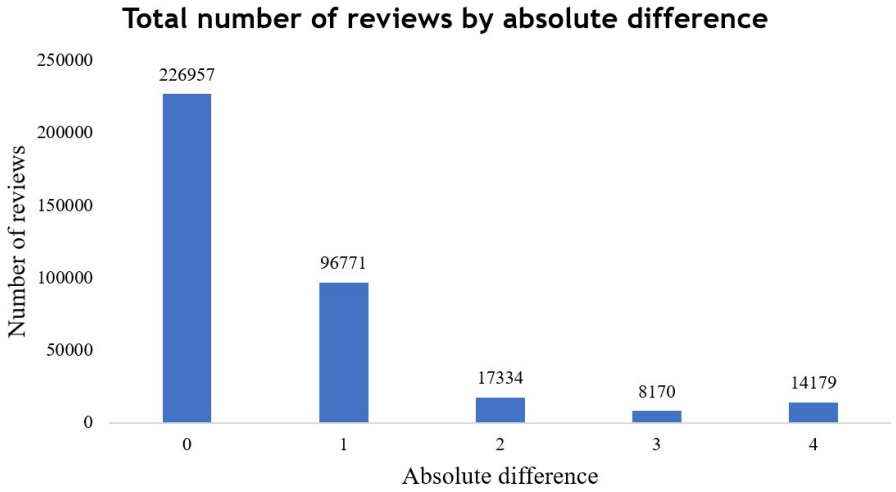


Figure 4.1: Total number of reviews by absolute difference

It was considered that a review with a minimum difference between the predicted textual rating and the current rating score of at least two stars is a review that should be corrected. In this note, the predicted classifier has poorly rated a total number of 39 683 reviews, representing 10,6% of the entire dataset before the pre-processing steps.
Furthermore, to test how the new textual rating is performing and the influence of this new predictor, a random sample of 150 products reviews from Worten's dataset that had a difference higher or equal to two were manually compared by five different people selecting if the new correction was a better interpretation of the review in terms of rating. The number of 150 samples was defined in order to make the participants' evaluation experience better, since an extremely high number

could lead to miss-classifications.
The results from this quick study show that despite the excellent accuracy obtained by the *XGBoost* classifier, only 43% of the corrections in a batch of 150 random samples were applicable. Furthermore, 39% of the corrections were considered negative corrections and 17% were inconclusive. Analyzing the random samples looking for why the negative corrections represented a significant number; it was found that one of the main reasons was the removal of negations in the data pre-processing step. However, it was not possible to quantify the impact of this problem.
The result seems acceptable even if the number of 43% of significant corrections sounds low, as it shows that, in reality, there are reviews that need to be corrected. Nevertheless, this result, combined with categorizing the reviews into business-related categories, can be used for deeper analysis.

4.2 Reviews Categorization

This subsection presents the results obtained using the Unsupervised Classification method, Zero-Shot Classification, used to classify reviews into six different categories for a more accessible analysis of the consumer's opinion on each critical point that supports a platform such as Worten.

4.2.1 Results

As it was previously noted in Section 3.6.1, a single textual product review can contain information regarding several aspects of the product and service provided by the company. This critical aspect makes classifying a review into a single category extremely difficult.
Figure 4.2 presents examples of the embedding results, in variable *sentence_embedding_bert*, obtained by using the Zero-Shot classification method with the *cross-encoder/nli-distilroberta-base* as an encoder.

sentence	sentence_embedding_bert
['deceiving', 'stock', 'customers']	[-0.28519004583358765, -0.14984790980815887, 0...
['already', 'see', 'item', 'times', 'say', 'av...	[-0.22075492143630981, 0.038028616458177567, 0...
['works', 'well', 'silent', 'combine', 'intere...	[-0.2708030641078949, -0.08995669335126877, 0....
['works', 'very', 'silent']	[-0.31470921635627747, -0.10875960439443588, 0...
['very', 'quiet', 'washes', 'well', 'fast', 'p...	[-0.372382789850235, -0.24486087262630463, 0.2...
...	...
['bought', 'product', 'week', 'extremely', 'sa...	[-0.5101334452629089, -0.12299250811338425, 0....
['be', 'satisfied', 'purchase', 'of', 'this', ...	[-0.376113623380661, -0.11843571811914444, 0.4...
['the', 'device', 'change', 'screen', 'correct...	[-0.42495352029800415, 0.051850516349077225, 0...
['screen', 'replacement']	[-0.23856432735919952, -0.1030733734369278, 0....
['according', 'to', 'intend']	[-0.2495376020669937, -0.13220621645450592, 0....

ns

Figure 4.2: Embedded sentences using the *cross-encoder/nli-distilroberta-base*

The results from this method were positive as a non-trained algorithm was able to obtain a total accuracy for the first level category of 69% in the Worten unsupervised dataset presented in Section 3.3.2. Furthermore, using the method previously described in Section 3.6.3 of introducing a second level to this classification the number of well-classified reviews obtaining an accuracy of 83%. However, this second method must be carefully interpreted as a new category level is introduced. The complete results obtained are presented in Table 4.4. Furthermore, the distribution for the first level predictions is presented in Figure 4.3.

Table 4.4: Results from the Zero-Shot classification method

Category	**Nº of previously labeled reviews**	**Nº of well-classified reviews in 1st level**	**Accuracy of 1st level**	**Nº of reviews rated at the 2nd level**	**Nº of well-classified reviews in the 2nd level**
Stock	50	23	46%	9	9
Product	50	44	88%	4	4
Price	50	41	82%	1	1
Description	50	24	48%	1	1
Delivery	50	32	64%	0	0
Marketplace	50	43	86%	2	2

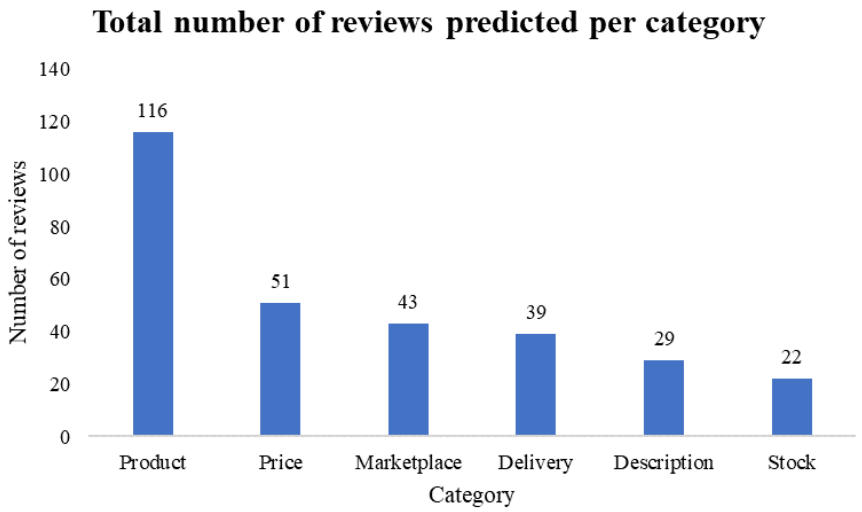


Figure 4.3: Distribution of the number of reviews predicted per category

Table 4.4 shows that the second-level category allocated for the reviews in the selected batch performed well as all the second-level categories associated correspond to the correct label.
Furthermore, a 2D spatial representation for the results of the S-BERT embedding for a batch of 5000 random reviews was developed using the method described in Section 3.6.2, Figure E.3 in Attachment E.

4.2.2 Discussion

Section 4.2.1 demonstrates that the Zero-Shot classification method can be used for classifying the reviews into six different categories as the method obtained a 69% accuracy in the first category level for the sample used, composed of 300 total manually pre-labeled reviews. Considering that this method has never been trained to perform this type of classification, the results are encouraging.
Analyzing Figure 4.3, it can be stated that the algorithm underperforms in classifying reviews related to *stock*, *description* of the products, and *delivery*. However, it performs well in the classes such as *marketplace*, *price*, and *product*. As seen in Figure 4.3, the Zero-Shot method classifies a vast number of reviews as product-related when in reality the reviews are from the other classes, as can be seen by the example that presents ten reviews with a second category level in Figure E.4 in Attachment E. The distribution of the predicted category for the entire Worten is presented in Table E.2 in Attachment E, where the variables in italic represent the first category level predicted and the variables in bold represent the second category level predicted.

4.3 Visualization results

This section is dedicated to describing the two dashboard prototypes developed that allow a better analysis and interpretation of the different reviews present in the company's platform and allow to analyze the performance of the two main tasks of this dissertation.

4.3.1 Development of the Power Bi dashboard

Two dashboard prototypes were developed: one to follow the results and the performance of the method used to create the new textual rating; another to show the reviews present in the company's platform categorized by theme.
The first dashboard developed presents different aspects of the predicted textual rating, and the objective of using this dashboard is to compare the textual rating predicted with the actual rating presented in the website. This dashboard is called *Overview of the algorithm predictions*. Firstly, the dashboard is composed of several filtering methods that allow the user to analyze the information presented at different levels:

- Firstly, the information can be filtered using the *SKU* number;
- Secondly, the information can be filtered by the first and second levels of the product category;
- Finally, and the most important one, the information can be filtered by the new textual rating predicted and the current website rating.

The combination of the different filtering methods allows the user to identify characteristics of the type of reviews that the *XGBoost* method classifies for each rating predicted and to know how each

product category performs in terms of rating.
For analyzing the reviews and the classifier's performance, a table is displayed that presents several variables such as the *SKU*, the *Review text*, the *Website Rating*, the *Textual Rating Predicted*, and the first and second-level *product category*. Furthermore, an additional table presents the average difference between the *Website's current Rating* and the *Textual Rating Predicted*. Finally, two graphics display the total number of reviews by difference and the average difference for each current *website rating*. Figure E.5 in Attachment E presents the dashboard proposed.
A quick and general analysis in this dashboard allowed us to conclude that the first-level product categories that had the higher average difference were the Home, the Perfumery, Cosmetics, and Beauty category, the Smarthome category, the Photography category, the Computers category, and the TV, Video and Sound category. Table E.3 in Attachment E presents the top ten categories where there is more difference. Furthermore, it will be interesting to further analyze the number of products sold by the company and the marketplace for each product category.
The second dashboard proposed presents the results obtained by the Zero-Shot classifier and aims to display the different reviews for each predicted category. This dashboard is called *Review categorization* and as the previous one, the dashboard is composed the several filtering methods:

- Using the SKU number;
- The first and second level of the product category;
- The predicted category;
- The current Website rating and the Textual rating Predicted.

Using the filters, the user can analyze each category's performance for a specific product or for a particular *product category* at a first and second level. Furthermore, filtering by a lower Textual rating Predicted, the user can extract information for problems related to each category predicted and for each product category. On the same note, the user can see the most valued aspects that consumers describe in reviews for the category filtering by higher ratings.
For analyzing the reviews a table displays multiple information such as the *SKU* of the product for which the review was written, the *Review text*, the first and second level *category predicted*, the first and second level of the *product category*, the *current website rating*, and the *Textual rating predicted*. Furthermore, one more table compares the average of the current Website rating and the Textual rating predicted for each category predicted. Finally, a graphic shows the total number of reviews for each predicted category by the current Website rating. Figure E.6 in Attachment E presents the second dashboard proposed.
Using the *Review categorization* dashboard, it was possible to identify two main predicted categories that are underperforming: the *Stock* and the *Delivery*. Reading the reviews in the table presented in the dashboard, it can be noted that when talking about the category stock, the leading consumer complaints are related to the lack of stock in products sold by Worten. The variable delivery is associated with the excessive time distribution companies take to deliver the product

and the fact that many packages arrive at the client's home damaged. On the other hand, the description on Worten's website is the category with a higher rating. Table E.4 in Attachment E presents the corresponding for each predicted category. Furthermore, analyzing both dashboards, it is interesting to see that the Textual Rating predicted has a higher rating than the current website rating for each category predicted.

Chapter 5

Conclusions and Future Work proposed

In this thesis, the focus was placed on evaluating several Machine Learning and Deep Learning approaches for text classification, with a particular interest in two main fields: assessing the potential of creating a new textual rating for reviews and categorizing reviews into six critical categories for any business that is related with e-commerce.

5.1 Conclusions

Firstly, a comprehensive assessment of the importance of the CRO (Conversion Rate Optimization) department and the importance of the consumer reviews on e-commerce platforms was conducted. The literature documents that e-WOM variables, such as the star rating and the textual element of a review, impact the sales volume of a product. Other variables such as the valence and the volume of reviews were also described as influencing the sales performance of a product. Furthermore, it was also documented that the CRO department impacted website retention and customer loyalty by dealing with several elements that increase customer satisfaction, such as ordering the products. Therefore, it is relevant to assess how the reviews can be used by the company as a method to retain consumers and how the company could use these reviews to detect problems related to several areas.

Several mismatches between the star rating and the textual element of the review were detected, causing products with fewer reviews to be automatically ranked at the bottom of the pages and, as such, to be less seen and bought by consumers as the overall rating as an impact on the ordering of the products. The idea of creating a new textual rating that could be compared with the current website rating was presented as a method to exhibit a platform with more trustful products in the first positions. This idea was reinforced when the number of products with fewer reviews on the platform was analyzed, especially in marketplace products. Moreover, it was detected that several aspects related to the company's performance were discussed in the reviews. Therefore, it was necessary to categorize the reviews into different categories so relevant information, such as problems, about each business aspect, could be extracted.

Secondly, a complete evaluation of the existing methods for text classification was conducted,

concluding that for the first objective, the SVM, Gradient Boosting and DL were the state-of-the-art Supervised Learning methods for text classification. Furthermore, the Zero-Shot classifier could be an Unsupervised Classification method for the second objective of the thesis. In addition, the comparison between the different Supervised Learning methods using three different datasets demonstrated that for the specific task of creating a new textual rating, the best classifier to use was the *XGBoost*.
The results of both classification methods were satisfying even if classifying text is usually complex, especially reviews that express opinions. On the one hand, the new textual rating could correct 43% on a batch of 150 reviews that had a significant difference between the current website rating and the new textual rating predicted. On the other hand, even if a review can contain multiple elements associated with different categories, the Zero-Shot classifier could correctly classify 69% of the reviews on a batch of 300 manually labeled reviews.
Finally, the need to better analyze the results and easily extract information motivated the creation of two dashboard prototypes in Power BI: *Overview of the algorithm predictions* and *Review categorization*. These prototypes allow to quickly analyze the predictions of the two classifiers used and extract some information about the different categories predicted. From these dashboards the main underperforming categories were noted: *Stock* and *Delivery*.
However, in the time-frame of this thesis, we were not able to implement neither the dashboards nor the new textual rating on the website, so it was not possible to measure the impact of the dissertation on the different CRO metrics such as CTR and Conversion Rate.

5.2 Future Work proposed

This section presents several possible future experiments, tests, and improvements that were not possible to develop throughout the duration of this dissertation. These future works proposed are related to the scope of this dissertation and concern the methods used for accomplishing both classification objectives and different possible uses and adaptations of the work developed.
Regarding the *Textual Rating*, some improvements and new ideas could be explored:

- Parameter tuning for the *XGBoost* classifier. As this method was the best classifier, it will be interesting to see if it is possible to increase its performance by doing parameter tuning in the parameters described in Table 3.7;

- Review the text pre-processing step. As previously described in Section 4.1.2, a problem was detected in the pre-processing step that might have influenced the final prediction; consequently, it will be interesting to find a way to make the information fed into the classifier more reliable;

- Develop a new DL structure. As the results from the DL method were unsatisfying, contradicting the different works presented in the literature, it will be interesting to develop several DL structures and test the true capabilities of DL;

- Measure the impact of implementing an automatic email or an internal warning that allows the worker of the company to send an email to the consumer later to see if the consumer wants to keep the rating of the review on customer behavior;
- Study the impact of implementing the textual rating next to the current rating and measure the impact on the CRO metrics with an AB test. As it was not possible to implement the new textual rating, it will be interesting and innovative to implement it on the website to see if it will impact the consumer's clicking and the conversion; however, the results from the classifier should be improved;
- Implement a feature extraction method capable of extracting product-related words from the reviews. It will be interesting to display these words next to the product image and the ratings and measure the impact on the CRO metrics with an AB test;
- Measure the impact of creating a reviewer rating on the number of reviews and their quality, in other words, the difference between the current website rating and the predicted textual rating. This reviewer rating should be related to the number and the quality of the reviews made by the consumer profile. The objective is to implement a promotions system for specific goals. Exploring the field of gamification, this solution will be attractive as the consumers will be interested in increasing the quality and the total number of reviews made.

Regarding the *Review categorization*, there are also some directions that could be further explored:

- Implement a Few-shot classification. Having obtained good results with the Zero-Shot classification, it will be interesting to test the Few-Shot classification method using meta-learning with the same dataset of 300 reviews used to evaluate the Zero-Shot method;
- Implement a feature extraction method that could extract relevant words from the filtered reviews displayed for the two dashboards to give a general idea of the consumer's sentiment and the main concerns to each product, product category, or category predicted.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zhen, X. (2015). Large-Scale Machine Learning on Heterogeneous Systems.

Adam Hayes (2021). What Is Stratified Random Sampling? `https://www.investopedia.com/terms/stratified_random_sampling.asp`. Accessed: 2022-05-10.

Afshine Amidi and Shervine Amidi (2020). Deep Learning cheatsheet. `https://stanford.edu/$\sim$shervine/teaching/cs-229/cheatsheet-deep-learning`. Accessed: 2022-05-24.

Al-Deen, H. S. S., Zeng, Z., Al-Sabri, R., and Hekmat, A. (2021). An improved model for analyzing textual sentiment based on a deep neural network using multi-head attention mechanism. *Applied System Innovation*, 4(4).

Alberto Pimenta (2021). CTT e-Commerce Report 2021. Technical Report November, CTT - Correios de Portugal, Direção de E-Commerce.

Alzamzami, F., Hoda, M., and Saddik, A. E. (2020). Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation. *IEEE Access*, 8:101840–101858.

Asogwa, T. C., Fidelis, E., Obodoeze, C., and Obiokafor, I. N. (2007). IJARCCE Wireless Sensor Network (WSN): Applications in Oil & Gas and Agriculture Industries in Nigeria. *International Journal of Advanced Research in Computer and Communication Engineering ISO*, 3297(1):156–159.

Azevedo, A. and Santos, M. F. (2008). KDD , SEMMA AND CRISP-DM : A PARALLEL OVERVIEW Ana Azevedo and M . F . Santos. *IADIS European Conference Data Mining*, pages 182–185.

Babenko, V. and Syniavska, O. (2018). Analysis of the current state of development of electronic commerce market in Ukraine. *Technology audit and production reserves*, 5(4(43)):40–45.

Barnden, J. A. (2008). Challenges in natural language processing: The case of metaphor (commentary). *International Journal of Speech Technology*, 11(3-4):121–123.

Bo Pang, L. L. and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Catelli, R., Pelosi, S., and Esposito, M. (2022). Lexicon-Based vs. Bert-Based Sentiment Analysis: A Comparative Study in Italian. *Electronics (Switzerland)*, 11(3).

Chakravarthy, V., Joshi, S., Ramakrishnan, G., Godbole, S., and Balakrishnan, S. (2008). Learning decision lists with known rules for text mining. *IJCNLP 2008 - 3rd International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2:835–840.

Chen, Q., Wang, W., Huang, K., and Coenen, F. (2021). Zero-shot Text Classification via Knowledge Graph Embedding for Social Media Data. *IEEE Internet of Things Journal*, pages 1–10.

Chen, T., He, T., and Benesty, M. (2018). XGBoost : eXtreme Gradient Boosting. *R package version 0.71-2*, pages 1–4.

Chen, Tianqi and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, San Francisco, California, USA.

Chevalier, J. A. and Mayzlin, D. (2006). The Effect of Word of Mouth on Sales: Online Book Reviews. *Journal of Marketing Research*, 45(2):345–354.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications, New York, United States, 2nd editio edition.

Christopher Olah (2015). Understanding LSTM Networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 2022-04-12.

Cowan, P. (2021). Meet Michael Aldrich, the godfather of online shopping. `https://www.smartosc.com/en/insights/michael-aldrich-godfather-online-shopping`. Accessed: 2022-04-13.

Cristina A. Ferreira (2021). Worten, Delta e Dott.pt aceleram a fundo nas vendas online e mantêm foco no digital em 2022. https://tek.sapo.pt/noticias/internet/artigos/worten-delta-e-dott-pt-aceleram-a-fundo-nas-vendas-online-e-mantem-foco-no-digital-em-2022. Accessed: 2022-04-23.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of the 12th International Conference on World Wide Web, WWW 2003*, (October 2003):519–528.

Davison, J. (2020). Zero-Shot Learning in Modern NLP. `https://joeddav.github.io/blog/2020/05/29/ZSL.html`. Accessed: 2022-04-26.

Duan, W., Gu, B., and Whinston, A. B. (2008). Do online reviews matter? - An empirical investigation of panel data. *Decision Support Systems*, 45(4):1007–1016.

EcommerceDB (2021). Store ranking of the e-commerce situation in Spain. `https://ecommercedb.com/en/ranking/es/all`. Accessed: 2022-05-10.

Edda, L. and Jörg, K. (2002). Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? *Machine Learning*, 46(1-3):423–444.

Fernández-Bonilla, F., Gijón, C., and De la Vega, B. (2022). E-commerce in Spain: Determining factors and the importance of the e-trust. *Telecommunications Policy*, 46(1).

Forman, C., Ghose, A., and Wiesenfeld, B. (2008). Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information Systems Research*, 19(3):291–313.

François Chollet et. al. (2015). Keras. `https://github.com/fchollet/keras`. Accessed: 2022-05-24.

Geng, R., Li, B., Li, Y., Zhu, X., Jian, P., and Sun, J. (2019). Induction networks for few-shot text classification. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3904–3913.

Gite, S., Khatavkar, H., Kotecha, K., Srivastava, S., Maheshwari, P., and Pandey, N. (2021). Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Computer Science*, 7:1–21.

Griva, A. (2022). "I can get no e-satisfaction". What analytics say? Evidence using satisfaction data from e-commerce. *Journal of Retailing and Consumer Services*, 66(July 2021).

Gunther, R. E. (2009). Peter Drucker-the grandfather of marketing: An interview with Dr. Philip Kotler. *Journal of the Academy of Marketing Science*, 37(1):17–19.

Guo, X., Zheng, S., Yu, Y., and Zhang, F. (2021). Optimal Bundling Strategy for a Retail Platform Under Agency Selling. *Production and Operations Management*, 30(7):2273–2284.

Hayat, Z., Rahim, A., Bashir, S., and Naeem, M. (2021). Self learning of news category using AI techniques. *Intelligent Environments 2021: Workshop Proceedings of the 17th International Conference on Intelligent Environments*, 29:167–178.

Hoch, R. (1994). Using IR techniques for text classification in document analysis. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, pages 31–40.

Hu, N., Liu, L., and Zhang, J. J. (2008). Do online reviews affect product sales? The role of reviewer characteristics and temporal effects. *Information Technology and Management*, 9(3):201–214.

Hu, N., Pavlou, P. A., and Zhang, J. (2006). Can online reviews reveal a product's true quality? pages 324–330.

Hu, Y., Jing, X., Ko, Y., and Rayz, J. T. (2020). Misspelling correction with pre-trained contextual language model. *Proceedings of 2020 IEEE 19th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2020*, pages 144–149.

IAB Spain (2021). Estudio eCommerce 2021. Technical report, IAB Spain. Accessed: 2022-05-13.

Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., Kolodziej, J., Wang, L., Chen, D., Rayes, A., Tziritas, N., Xu, C. Z., Zomaya, A. Y., Alzahrani, A. S., and Li, H. (2015). A survey on text mining in social networks. *Knowledge Engineering Review*, 30(2):157–170.

Jagdale, R. S., Shirsat, V. S., and Deshmukh, S. N. (2019). *Sentiment analysis on product reviews using machine learning techniques*, volume 768. Springer Singapore.

Jiawei Han (2014). *Data mining: concepts and techniques*. pages 607–618.

Josh Steimle (2015). What Is Conversion Rate Optimization? https://www.forbes.com/sites/joshsteimle/2015/07/14/what-is-conversion-rate-optimization/?sh=1d7c9cbd6a0f. Accessed: 2022-04-21.

Khurana, D., Koli, A., Khatter, K., and Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges. (Figure 1).

Ko, Y. and Seo, J. (2000). Automatic text categorization by unsupervised learning. pages 453–459.

Kostyra, D. S., Reiner, J., Natter, M., and Klapper, D. (2016). Decomposing the effects of online customer reviews on brand, price, and product attributes. *International Journal of Research in Marketing*, 33(1):11–26.

Lee, C. H. and Yang, H. C. (2009). Construction of supervised and unsupervised learning systems for multilingual text categorization. *Expert Systems with Applications*, 36(2 PART 1):2400–2410.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *Ecml*, volume 1398, pages 4–15. Springer, Berlin, Heidelberg.

Li, H. and Yamanishi, K. (2002). Text classification using ESC-based stochastic decision lists. *Information Processing and Management*, 38(3):343–361.

Ma, C., Zhang, W. E., Guo, M., Wang, H., and Sheng, Q. Z. (2020). Multi-document Summarization via Deep Learning Techniques: A Survey. 1(1):1–35.

Ma, T., Yao, J. G., Lin, C. Y., and Zhao, T. (2021). Issues with Entailment-based Zero-shot Text Classification. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2:786–796.

Malmberg, J. (2021). Evaluating semantic similarity using sentence embeddings.

Maloney, D., Freeman, G., and Wohn, D. Y. (2020). "Talking without a Voice": Understanding Non-Verbal Communication in Social Virtual Reality. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2).

Marktest, G. (2021). Ranking de sites e-Commerce de junho de 2021. https://www.marktest.com/wap/a/n/id$\sim$27b2.aspx. Accessed: 2022-06-05.

Masui, T. (2020). All You Need to Know about Gradient Boosting Algorithm. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502. Accessed: 2022-05-15.

Miikkulainen, R., Iscoe, N., Shagrin, A., Cordell, R., Nazari, S., Schoolland, C., Brundage, M., Epstein, J., Dean, R., and Lamba, G. (2017). Conversion rate optimization through evolutionary computation. *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, pages 1193–1199.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2020). Deep Learning Based Text Classification: A Comprehensive Review. 1(1):1–43.

Moe, W. W., Trusov, M., and Smith, R. H. (2011). The value of social dynamics in online product ratings forums. *Journal of Marketing Research*, 48(3):444–456.

Nadkarni, P. M., Ohno-Machado, L., and Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551.

Nagesh Singh Chauhan (2021). Zero-Shot Learning: Can you classify an object without seeing it before? `https://www.kdnuggets.com/2021/04/zero-shot-learning.html`. Accessed: 2022-05-23.

Nahm, U. and Mooney, R. (2002). Text mining with information extraction. *Spring Symposium on Mining Answers from Texts and Knowledge Bases*, (September 2003):1–16.

Nikolay, A., Anindya, G., and Panagiotis, G. I. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management Science*, 57(8):1485–1509.

Nils Reimers (2021). Cross-Encoder for Natural Language Inference. `https://huggingface.co/cross-encoder/nli-distilroberta-base/tree/main`. Accessed: 2022-05-10.

Parveen, R., Shrivastava, N., and Tripathi, P. (2013). Sentiment classification of movie reviews by supervised machine learning approaches voted algorithm. *Indian Journal of Computer Science and Engineering (IJCSE) SENTIMENT*, pages 285–292.

Pathak, R. and Thankachan, B. (2012). Natural language processing approaches, application and limitations. *International Journal of Engineering Research & Technology*, 1(7):1–8.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duche, E. (2011). Scikit-learn: Machine Learning in Python.

Perez Amaral, T., Valarezo Unda, A., Lopez, R., Garín-Muñoz, T., and Herguera García, I. (2019). E-Commerce and Digital Divide in Spain Using Individual Panel Data 2008-2016. *SSRN Electronic Journal*, pages 2016–2018.

Pupale, R. (2018). Support Vector Machines(SVM) — An Overview. `https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989`. Accessed: 2022-05-10.

Raghupathi, D., Yannou, B., Farel, R., and Poirson, E. (2014). Sentiment rating algorithm of product online reviews. *Proceedings of International Design Conference, DESIGN*, 2014-Janua:2135–2146.

Rajpoot, A. K., Nand, P., and Abidi, A. I. (2021). Development of textual analysis using machine learning to improve the sentiment classification. *Journal of Physics: Conference Series*, 2062(1):0–7.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3982–3992.

Rita, P., Oliveira, T., and Farisa, A. (2019). The impact of e-service quality and customer satisfaction on customer behavior in online shopping. *Heliyon*, 5(10):e02690.

Rokach, L. and Maimon, O. (2008). Decision Trees. *Lecture Notes in Mathematics*, 1928:67–86.

Saleem, H., Khawaja, M., Uddin, S., Habib-Ur-Rehman, S., Saleem, S., and Aslam, A. M. (2019). Strategic Data Driven Approach to Improve Conversion Rates and Sales Performance of E-Commerce Websites. *International Journal of Scientific & Engineering Research*, 10(4):588–593.

Sassano, M. (2003). Virtual examples for text classification with Support Vector Machines. (1995):208–215.

Schröer, C., Kruse, F., and Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, 181(2019):526–534.

Sebastian Poliak (2020). 1 to 5 Star Ratings — Classification or Regression? `https://towardsdatascience.com/1-to-5-star-ratings-classification-or-regression-b0462708a4df`. Accessed: 2022-05-20.

Shafiabady, N., Lee, L. H., Rajkumar, R., Kallimani, V. P., Akram, N. A., and Isa, D. (2016). Using unsupervised clustering approach to train the Support Vector Machine for text classification. *Neurocomputing*, 211:4–10.

Shi, D., Wang, M., and Li, X. (2021). Strategic introduction of marketplace platform and its impacts on supply chain. *International Journal of Production Economics*, 242(September):108300.

Sourav, D., Sandip, D., Siddhartha, B., and Surbhi, B. (2022). *Advanced Data Mining Tools and Methods for Social Computing*. Academic Press, 1st editio edition.

Stecanella, B. (2019). Understanding TF-ID: A Simple Introduction. `https://monkeylearn.com/blog/what-is-tf-idf/`. Accessed: 2022-05-12.

Tripathy, A., Agrawal, A., and Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques. *Procedia Computer Science*, 57:821–829.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009.

Wei, Y. and Dong, Y. (2022). Product distribution strategy in response to the platform retailer's marketplace introduction. *European Journal of Operational Research*, (xxxx).

Weiscbedel, R., Carbonell, J., Grosz, B., Lehnert, W., Marcus, M., Perrault, R., and Wilensky, R. (1989). White Paper on Natural Language Processing. *Speech and Natural Language, Proceedings of a Workshop*, pages 481–493.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. *International Conference on Information and Knowledge Management, Proceedings*, pages 625–631.

Wigand, R. T. (1997). Electronic commerce: Definition, theory, and context. *Information Society*, 13(1):1–16.

Yechuri, P. K. and Ramadass, S. (2021). Classification of Image and Text Data Using Deep Learning-Based LSTM Model. *Traitement du Signal*, 38(6):1809–1817.

Zhang, D. J., Dai, H., Dong, L., Qi, F., Zhang, N., Liu, X., Liu, Z., and Yang, J. (2020). The long-term and spillover effects of price promotions on retailing platforms: Evidence from a large randomized experiment on alibaba. *Management Science*, 66(6):2589–2609.

Zhang, P., Zhou, L., and Zimmermann, H. D. (2013). Advances in social commerce research: Guest editors' introduction. *Electronic Commerce Research and Applications*, 12(4):221–223.

Zhang, S. and Zhang, J. (2020). Agency selling or reselling: E-tailer information sharing with supplier offline entry. *European Journal of Operational Research*, 280(1):134–151.

Appendix A

Models Description

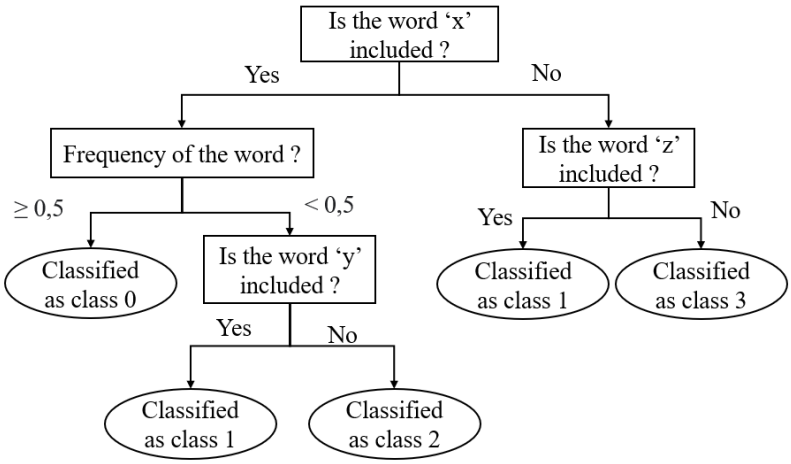
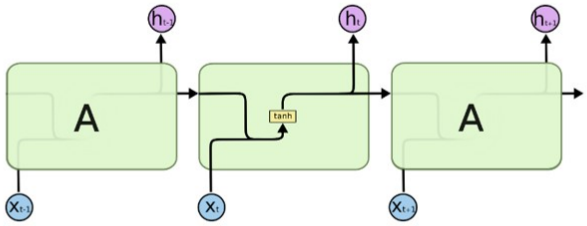


Figure A.1: Theorical example of a decision tree adapted to a text classification problem

The cell state is the key to LSTMs represented by the horizontal line running through the different cells. This cell can be seen as a conveyor belt that transports information running through the entire chain that is subjected to interactions.



This structure allows LSTMs the ability to add or remove information to the cell state. This action is regulated by gates. As the name suggests, gates regulate the information that passes through by applying multiplication operations in a defined neural network layer, in the case of the image presented, the activation function used is the tanh.

Figure A.2: *LSTM* structure. Adapted from Christopher Olah (2015)

Appendix B

Data extracted from Worten's database

Id_Review	Review Submission Date	Review Display Locale	Campaign ID	Moderation Status	Moderation Codes	Ratings-only (Y/N)	Overall Rating	Review Title	Review Text	Product ID	Product Name	Product Page URL	Brand	Category Hierarchy	Reviewer ID
1030401879	2017-03-23 20:36:26	pt_PT	BV_REVIEW_DISPLAY	APPROVED	HMP	No	4.0	Boa relaÃ§Ã£o qualidade/preÃ§o	Boa relaÃ§Ã£o qualidade/preÃ§o e com um Ã³timo...	1006077345	MÃ¡quina de Lavar Roupa SAMSUNG WF80F5E0W2W (8...	https://www.worten.pt/grandes-eletrodomesticos...	SAMSUNG	Grandes EletrodomÃ©sticos > MÃ¡quinas de Roupa...	qehpcfi6gfzusmraarldgzibj
1030400159	2017-03-23 18:43:42	pt_PT	BV_REVIEW_DISPLAY	APPROVED	HMP	No	4.0	Boa relaÃ§Ã£o qualidade / preÃ§o	MÃ¡quina bastante funcional e com alta qualida...	1030667435	Action cam ROLLEI AC425 WiFi 4K (4K - 12 MP - ...	https://www.worten.pt/ultimas-unidades/action-...	ROLLEI	Ãltimas Unidades	tswlboaew1iiltagaavbqzkmx
1030399977	2017-03-23 18:31:06	pt_PT	BV_REVIEW_DISPLAY	APPROVED	HMP	No	5.0	Pulseira Fit e mais alÃ©m	Comprei este produto porque descobri que tinha...	1030639803	Pulseira Desportiva SAMSUNG Geat Fit2 (Bluetoo...	https://www.worten.pt/ultimas-unidades/pulseir...	SAMSUNG	Ãltimas Unidades	tswlboaew1iiltagaavbqzkmx
1030399886	2017-03-23 18:24:23	pt_PT	BV_REVIEW_DISPLAY	APPROVED	HMP	No	5.0	excelente comprei a 15 dias estou satisfeito	produto muito bom e a bom preÃ§o de excelente ...	1006073926	Micro-ondas SAMSUNG Mc28h5135ck Preto	https://www.worten.pt/pequenos-eletrodomestico...	SAMSUNG	Pequenos EletrodomÃ©sticos > Micro-ondas e Min...	mk60b22baobcxlalc7rklidqf
1029392229	2017-03-02 10:03:04	NaN	BV_REVIEW_DISPLAY	PURGED	RTF	No	NaN	NaN	NaN	1030665790	TV LED 32'' SHARP LC-32CHE4042EH	https://www.worten.pt/tv-video-e-som/tvs/peque...	SHARP	TV, VÃdeo e Som > TVs > TV Pequena Polegada	qfksg433wjlwf0saboka0ik7j

Figure B.1: Sample of the dataset extracted from Worten's database

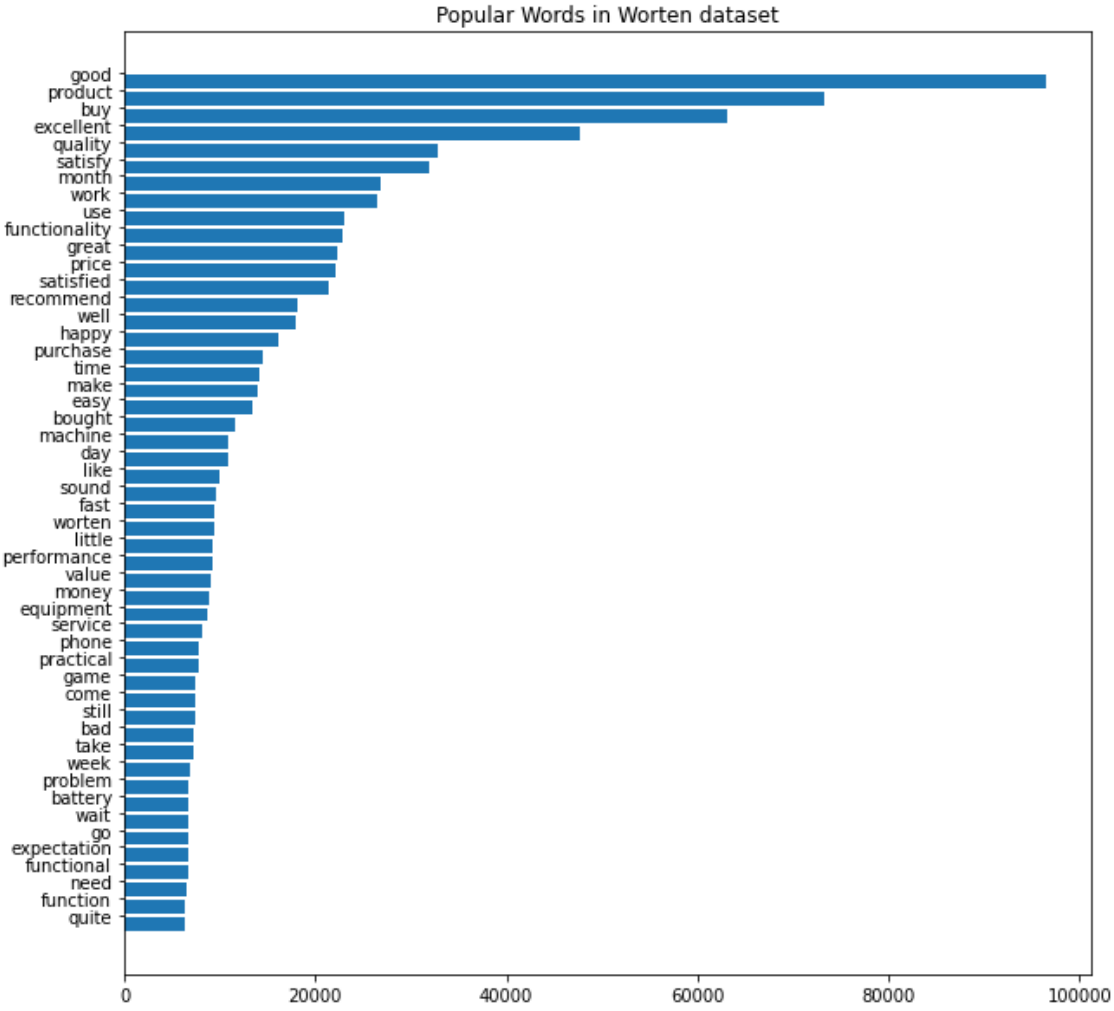


Figure B.2: Most frequent words of Worten's dataset

Appendix C

Data Preparation

Table C.1: Distribution of the train, test and validation set for each dataset

Dataset	Rating	Train	Test	Validation
Worten	*1*	4000	500	500
	2	2000	250	250
	3	2000	250	250
	4	12000	1500	1500
	5	40000	5000	5000
Amazon	*1*	4000	500	500
	2	2000	250	250
	3	6000	750	750
	4	8000	1000	1000
	5	40000	5000	5000
Coursera	*1*	1600	200	200
	2	1600	200	200
	3	3200	400	400
	4	9600	1200	1200
	5	32000	4000	4000

Table C.2: Stop words list from *NLTK* stop words library

i	me	my	myself	we	our	ours	wasn
ourselves	you	youre	youve	youll	youd	your	shant
yours	yourself	yourselves	he	him	his	himself	shouldn
she	shes	her	hers	herself	it	its	shouldnt
its	itself	they	them	their	theirs	themselves	mightn
what	which	who	whom	this	that	thatll	mightnt
these	those	am	is	are	was	were	mustn
be	been	being	have	has	had	having	needn
do	does	did	doing	a	an	the	mustnt
and	but	if	or	because	as	until	neednt
while	of	at	by	for	with	about	shan
against	between	into	through	during	before	after	wont
above	below	to	from	up	down	in	wasnt
out	on	off	over	under	again	further	weren
then	once	here	there	when	where	why	werent
how	all	any	both	each	few	more	won
most	other	some	such	no	nor	not	wouldnt
only	own	same	so	than	too	very	wouldn
s	t	can	will	just	don	dont	ma
should	shouldve	now	d	ll	m	o	isnt
re	ve	y	ain	aren	arent	couldn	isn
couldnt	didn	didnt	doesn	doesnt	hadn	hadnt	havent
hasn	hasnt	haven					

Appendix D

Models used

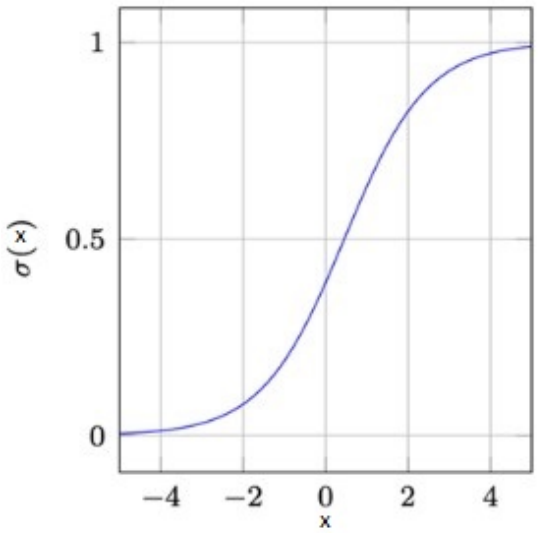


Figure D.1: *Softmax* Activation function representation. Adapted from Chollet (2017)

Table D.1: Description of the parameters used in each layer of the DL model. Adapted from François Chollet et. al. (2015)

Layer	Parameter	Function	Value
Embedding	*input_dim*	Size of the vocabulary	150000
	output_dim	Dimension of the dense embedding	360
	embeddings_ini	Initializer for the embedding's matrix	Uniform
	embeddings_reg	Regularizer function applied to the embedding's matrix	None
	embeddings_const	Constraint function applied to the embedding's matrix	None
	mask_zero	Special "padding" value that should be masked out	True
	input_length	Length of input sequences	100
Conv1D	*filters*	Dimensionality of the output space	10
	kernel_size	Specifying the length of the 1D convolution window	5
	strides	Specifying the stride length of the convolution	None
	padding	Padding with zeros evenly to the left/right or up/down of the input such that output has the same height/width dimension	Same
	activation	Activation function to use	Sotfmax
	use_bias	Whether the layer uses a bias vector	False
	kernel_initializer	Initializer for the kernel weights matrix	glorot_uniform
LeakyReLU	*alpha*	Negative slope coefficient	0.3
MaxPooling1D	*pool_size*	Size of the max pooling window	5
	strides	Specifies how much the pooling window moves for each pooling step	None
LSTM	*units*	Dimensionality of the output space	32
	activation	Activation function to use	Tanh
	recurrent_activation	Activation function to use for the recurrent step	Sigmoid
	dropout	Fraction of the units to drop for the linear transformation of the inputs	0.5
	recurrent_dropout	Fraction of the units to drop for the linear transformation of the recurrent state	0
	return_sequences	Whether to return the last state in addition to the output	True
	unroll	The network will be unrolled	False
	use_bias	Whether the layer uses a bias vector	False
Dropout_2	*rate*	Fraction of the input units to drop	0.3
	seed	Random state	random
Dense_1	*units*	Dimensionality of the output space	10
	activation	Activation function to use	Softmax
	kernel_initializer	Initializer for the kernel weights matrix	glorot_uniform
Dropout_3	*rate*	Fraction of the input units to drop	0.2
	seed	Random state	random
Dense_2	*units*	Dimensionality of the output space	5
	activation	Activation function to use	Softmax
	kernel_initializer	Initializer for the kernel weights matrix	glorot_uniform

Appendix E

Results

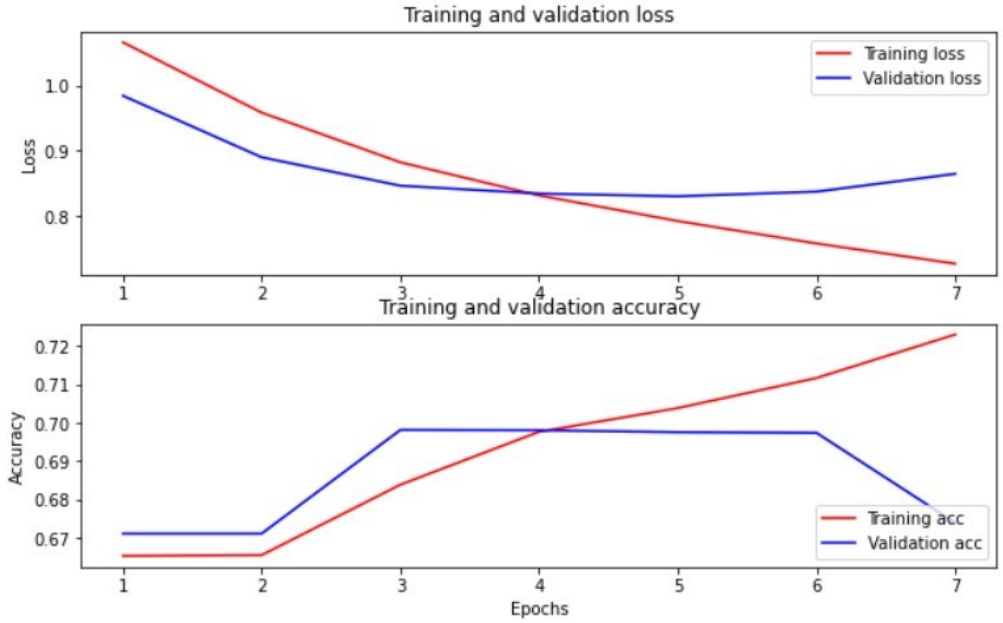


Figure E.1: Training and Validation Loss and Accuracy for 7 epochs overfitting in the training loop

Table E.1: Confusion matrix for the DL model (the labels presented in each row represent the true label of the text and the columns represent the predicted labels)

	1	**2**	**3**	**4**	**5**
1	2770	0	0	0	2230
2	1158	0	0	0	1342
3	402	0	0	0	2098
4	371	0	0	0	14629
5	770	0	0	0	49230

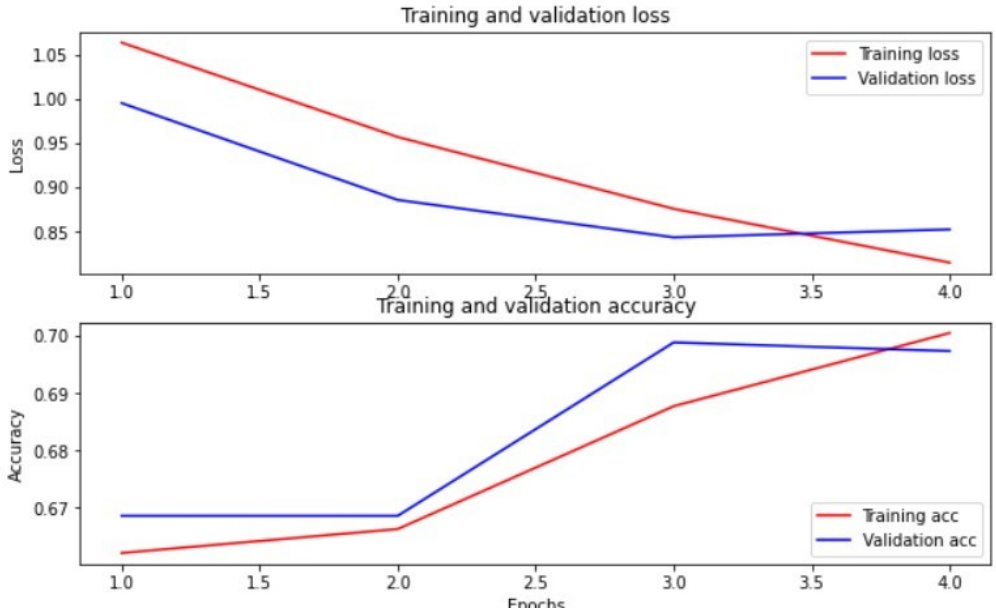


Figure E.2: Training and Validation Loss and Accuracy for 4 epochs without overfitting in the training loop

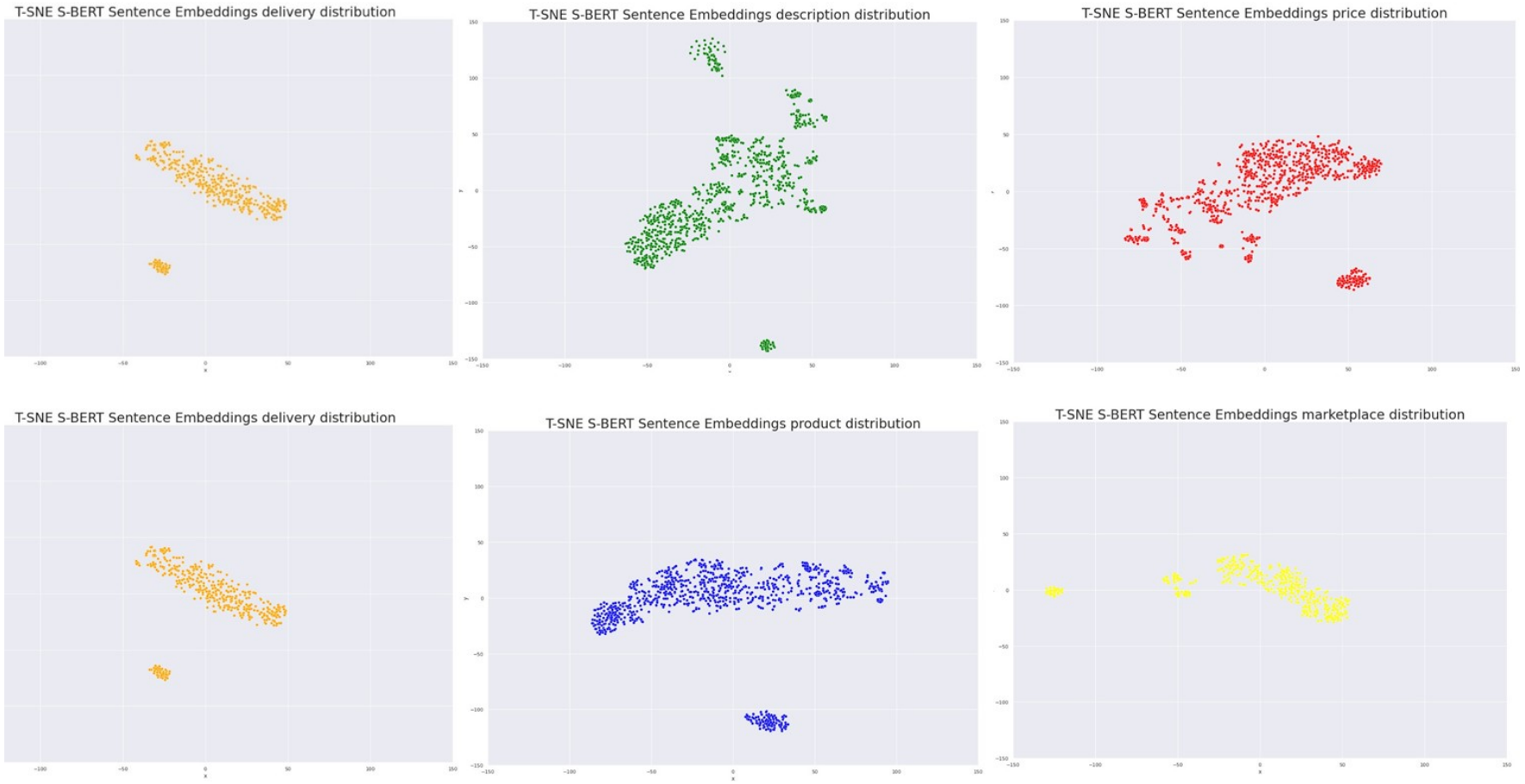


Figure E.3: 2D spatial representation for the results of the S-BERT embedding

Text_final_do_dataset	first_cat	second_cat
buy mobile phone spend almost month cont...	product	stock
buy iphone worten online available make ...	product	stock
seller accept purchase last week issue r...	product	stock
fact im still wait stupid always leave ...	product	stock
day make purchase receive message say eq...	product	stock
false product available month ago wait o...	product	stock
stupidity worten day end campaign say pr...	product	marketplace
recommend worten recommend supplier keru ...	product	marketplace
original cost little exist market product...	product	price
board unavailable company worten know pk ...	product	description

Figure E.4: Results where the first predicted category is product, and a second category was defined

Table E.2: Number of reviews in each predicted category for the first and second level

Category	**Stock**	**Description**	**Price**	**Delivery**	**Product**	**Marketplace**	**No second level**
stock	0	259	15	13	136	41	293
description	632	0	6573	1670	101645	2271	66159
price	21	4255	0	98	7798	2269	26178
delivery	10	984	109	0	705	76	3668
product	153	40615	5255	829	0	3664	80369
marketplace	12	1736	627	90	2083	0	2100

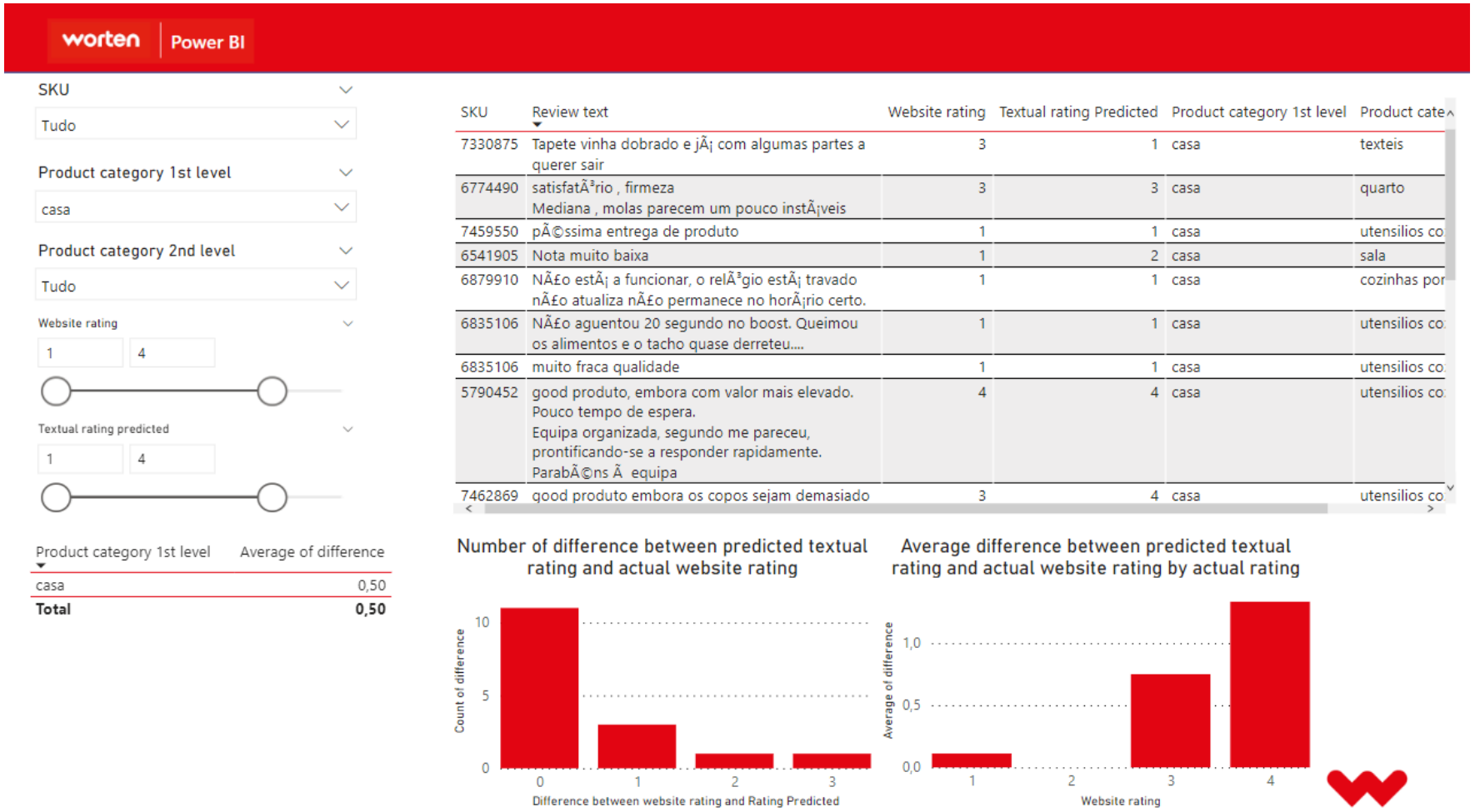


Figure E.5: Prototype for the dashboard: *Overview of the algorithm predictions*

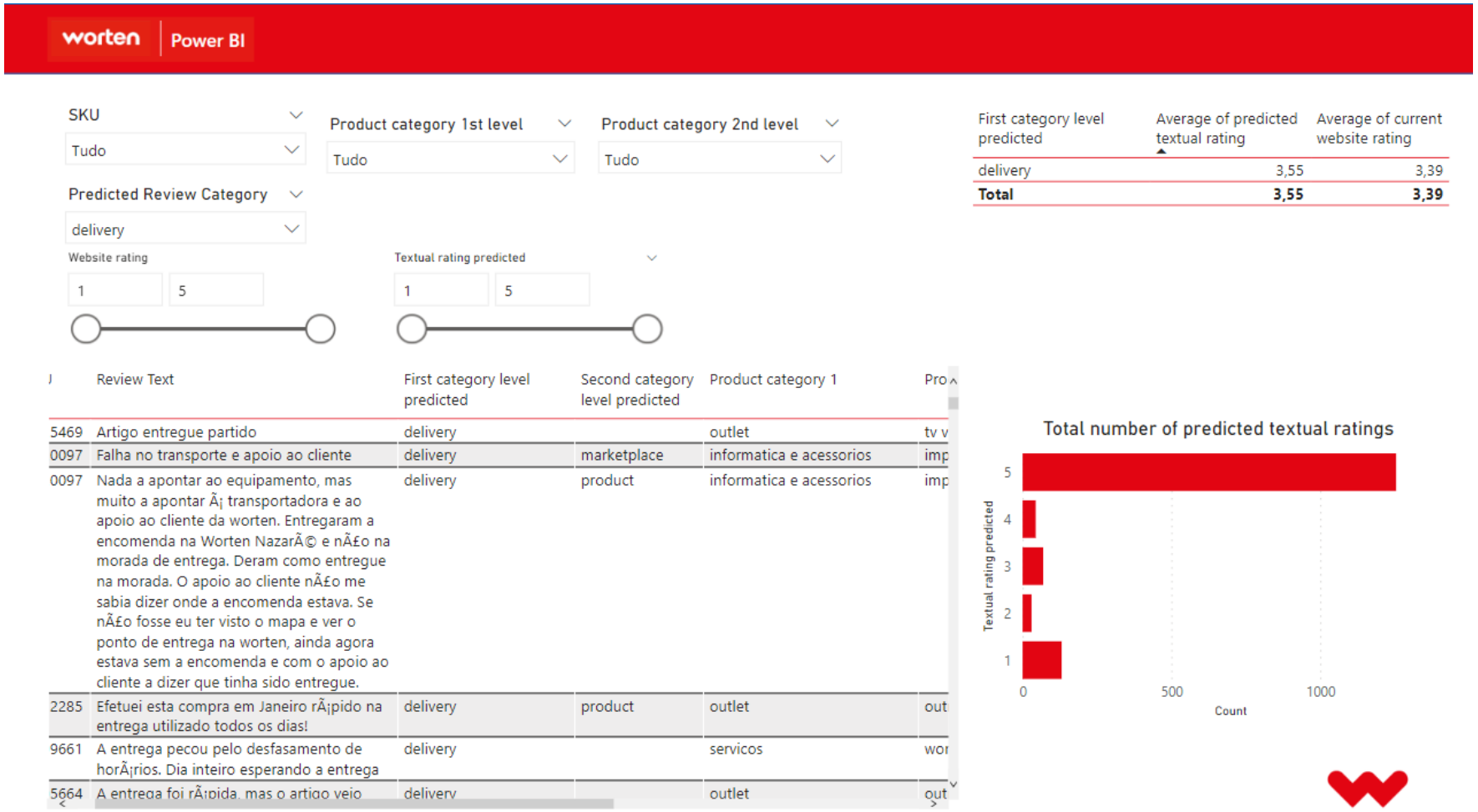


Figure E.6: Prototype for the dashboard: *Review categorization*

Table E.3: Top 10 product categories with more difference between the Textual Rating predicted and the current Website Rating

Product category	**Average difference**
Home	0,5
Perfumery, Cosmetics, and Beauty	0,33
Smarthome	0,28
Photography	0,28
Computers	0,21
TV, Video and Sound	0,19
Smartphones	0,18
Small appliances	0,18
Office	0,17
Gift cards	0,17

Table E.4: Comparison between the Textual Rating predicted and the current Website rating for each category predicted

1st level category predicted	**Avg. Textual Rating predicted**	**Avg. current Website rating**
Stock	3,52	3,19
Delivery	3,55	3,39
Product	4,24	3,95
Marketplace	4,33	3,98
Price	4,46	4,17
Description	4,48	4,17