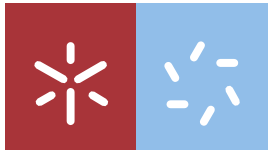
Sentiment Analysis – A Machine Learning Approach to Improve Customer Service

Catarina Campos Almeida

UMinho | 2023

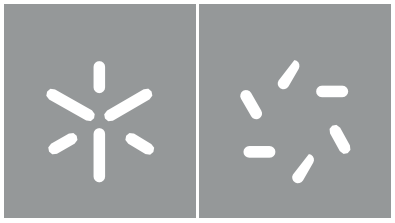


Universidade do Minho
Escola de Ciências

Catarina Campos Almeida

Sentiment Analysis – A Machine Learning Approach to Improve Customer Service

july 2023



Universidade do Minho
Escola de Ciências

Catarina Campos Almeida

Sentiment Analysis – A Machine Learning Approach to Improve Customer Service

Master's Dissertation
Master in Mathematics and
Computer Science

Work carried out under the guidance of

Cecília Castro

Ana Freitas

july 2023

Acknowledgements

First of all, I would like to express my gratitude to my advisor, Cecília Castro, for her unwavering dedication and availability throughout this project. I sincerely thank her for all the encouragement and opportunities she provided me, always aiming to help me improve. Thank you for all the contributions that undoubtedly enriched this journey.

I would also like to give special thanks to Liliana Bernardino and Ana Freitas for the great opportunity they gave me to be part of their team and for the opportunity to work on this project; thank you for everything.

To my team, Ana, Filipe, Catarina, Alexandre, Nuno, Carlos and Rodrigo, my thanks for the support since the beginning of the project and for all that I learned with you.

Last but not least, I want to thank my parents and my brother Miguel for being my greatest example of resilience, determination, and above all, love. One last thank you to all my friends and boyfriend for being such an important part of my life.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

”Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.” - Marie Curie

Abstract

Companies currently have a large amount of textual data coming from e-mails and/or forms sent by their customers. The cost associated with the manual response to each customer is very high due to the massive amount of this type of information. It is, therefore, in this sense of supporting the customer support team of a portuguese retailer that this dissertation aims to contribute with an artificial intelligence model capable of classifying the sentiment present in the messages sent by customers, as well as with tools that allow the identification of the most present themes in the messages that reach the team, from the most negative to the most positive feedbacks.

For this, this dissertation addresses text processing techniques applied to sentiment analysis, such as removing words that do not contribute significantly to the identification of sentiment, known as stopwords and eliminating characters that may interfere with the proper interpretation of the text.

In addition to text processing techniques, this dissertation presents a detailed analysis of several models used in Sentiment Analysis. Traditional methods such as naive Bayes, random forests, logistic regression, XGBoost, and ordinal regression are explored, as well as approaches based on pre-trained models.

This dissertation resulted in the development of a machine learning model of sentiment analysis. The model has a high ability to identify the sentiment present in texts, which can assist the customer service team in responding to customers in a personalized way, providing a more effective and satisfactory service.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Retail

Resumo

Atualmente, as empresas dispõem de uma grande quantidade de dados textuais provenientes de e-mails e/ou formulários enviados pelos seus clientes. O custo associado à resposta manual a cada cliente é muito elevado devido à quantidade massiva deste tipo de informação. É, então, neste sentido de apoiar a equipa de apoio ao cliente de um retalhista português que esta dissertação tem como objetivo contribuir com um modelo de inteligência artificial capaz de classificar o sentimento presente nas mensagens enviadas pelos clientes, bem como com ferramentas que permitam a identificação dos temas mais presentes nas mensagens que chegam à equipa, desde os *feedbacks* mais negativos aos mais positivos.

Para isto, esta dissertação aborda técnicas de processamento de texto aplicadas à análise de sentimento, como por exemplo remoção de palavras que não contribuem significativamente para a identificação de sentimentos, conhecidas como *stopwords* e eliminação de caracteres que possam interferir na interpretação adequada do texto.

Além das técnicas de tratamento de texto, esta dissertação apresenta uma análise detalhada de diversos modelos utilizados em análise de sentimento. São explorados métodos tradicionais, como *naive Bayes*, *random forests*, *logistic regression*, *XGBoost* e *ordinal regression*, bem como abordagens baseadas em modelos pré-treinados.

Esta dissertação resultou no desenvolvimento de um modelo de aprendizagem automática de análise de sentimento. O modelo possui uma alta capacidade de identificação do sentimento presente em textos, o que pode auxiliar a equipa de apoio ao cliente na resposta personalizada aos clientes, proporcionando um atendimento mais eficaz e satisfatório.

Palavras-chave: Análise de Sentimento, Processamento de Linguagem Natural, Aprendizagem Automática, Retalho

Contents

List of Figures

List of Tables

Acronyms and Symbols

SA - Sentiment Analysis
CS - Customer Service
NLP - Natural Languace Processing
NB - Naive Bayes
MLR - Multinomial Logistic Regression
OR - Ordinal Regression
SVM - Support Vector Machines
RF - Random Forests
XGBoost - eXtreme Gradient Boost
LRT - Likelihood Ratio Test
ROC - Receiver Operating Characteristic
AUC - Area Under the ROC Curve
NB - Naive Bayes
TP - True Positive
FN - False Negative
FP - False Positive
TN - True Negative
SW - Stop Words
SS - Sentiment Score
PS - Positive Score
NS - Negative Score
NA - Number of Adjectives
LD - Lexical Diversity

Chapter 1

Introduction

1.1 Introduction to Sentiment Analysis

Sentiment analysis is a computational process of identifying and categorizing sentiments present in sentences and text, determining whether a reviewer's position on a specific issue or product is positive, negative, or neutral. With the huge growth of digital platforms such as discussion forums, blogs, etc., the number of opinions shared on the web is also growing. Hence, it becomes impossible for stakeholders (entrepreneurs, politicians, etc.) to read and sort through thousands of comments in a timely manner. Thus, sentiment analysis has become an essential tool for understanding and extracting valuable information from large volumes of text.

The ability to automatically analyze sentiment has numerous applications in various industries. In customer service, sentiment analysis enables companies to identify and respond promptly to customer concerns, improving their overall customer experience. In addition, sentiment analysis plays a crucial role in reputation management, brand monitoring and market research, providing organizations with valuable information about their online presence and consumer sentiment.

1.2 The Importance of Sentiment Analysis in Retail

In today's competitive retail reality, understanding and responding effectively to customers by taking their feelings into account is of huge importance to businesses. The rise of social media, online reviews and customer feedback platforms has given retailers access to vast amounts of text data that can be analyzed to understand customer opinions and sentiments. This is where sentiment analysis plays a crucial role.

SA in retail involves the computational analysis of customer sentiments expressed in text, such as product and service reviews. By using NLP and machine learning techniques, sentiment analysis can automatically classify customer sentiments as positive, negative or neutral, providing valuable information about customer experiences and perceptions.

One of the main reasons for the ever growing importance of sentiment analysis in retail is the power of customer opinions and recommendations. With the spread of e-commerce platforms, consumers rely heavily on online ratings and reviews to make purchasing decisions. Positive reviews can significantly influence potential customers, driving sales and improving brand reputation. On the other hand, negative reviews can deter customers, leading to lost sales and potential brand image damage. Sentiment analysis allows retailers to systematically analyze and aggregate customer comments, gaining a comprehensive understanding of the overall sentiment towards their products and services.

1.3 Project Objectives

In the context of supporting the customer service team of a portuguese retailer, the primary objective of this dissertation is to develop a robust artificial intelligence model that can effectively classify the sentiment expressed in the messages received from customers. The model will leverage advanced natural language processing techniques to accurately analyze the content of these messages and determine whether the sentiment is positive, negative, or neutral.

Additionally, this research aims to provide the customer service team with powerful tools that facilitate the identification of prevalent themes in the incoming messages. By implementing text mining techniques, the team will be able to gain insights into the most frequently reported subjects, enabling them to better understand customer concerns and preferences. By achieving these objectives, the dissertation aims to enhance the overall efficiency and effectiveness of the CS team.

The application of artificial intelligence and advanced text analytics techniques will enable the team to promptly identify and prioritize customer issues, as well as automate responses, resulting in improved customer satisfaction, increased operational efficiency, and fortified brand loyalty.

1.4 MC Sonae and its Loyalty Program

MC Sonae operates a diverse range of food-based stores, including hypermarkets, large supermarkets, and small supermarkets. Each store format srves to different customer segments and offers a varied selection of products. To further enhance customer loyalty and gain insights into individual shopping habits, MC Sonae launched Cartão Continente on January 23, 2007, that currently includes 19 permanent partners and some occasional partners in several areas such as food, transportation, health, and fashion. This loyalty card initiative currently boasts approximately 4 million active accounts.

Cartão Continente not only serves as a means to increase brand loyalty but also provides MC Sonae with valuable data on consumer behavior at an individual level. By analyzing the shopping patterns of customers, the company can effectively manage its product range, make informed decisions regarding store expansions and adaptations, drive product innovation, and develop customer-focused strategies and loyalty programs. With this, the company has valuable information that can be used to individualize and personalize customer responses, taking into account the sentiment attributed to each customer.

1.5 Dissertation Structure

This dissertation is structured into five main sections, each focusing on a specific aspect of the research on sentiment analysis and its application in the retail industry. The following provides an overview of the content covered in each section.

First, in the Introduction section, the background of the dissertation is presented. It begins with an introduction to sentiment analysis, highlighting its importance in understanding customer feedback and opinions. The importance of sentiment analysis, specifically in the retail sector, is also highlighted. The section continues with a presentation of the project objectives, outlining the goals and intentions of the research and, finally, ends with a topic that introduces MC Sonae and its loyalty program.

The Literature Review section explores into existing research and scholarly work related to sentiment analysis. It explores various approaches and techniques used in sentiment analysis, with a focus on machine learning and lexicon-based methods. This section provides a comprehensive understanding of the current state of the field, highlighting the strengths and limitations of different approaches.

The next section describes the Methodology used in the research. It begins by

presenting the different models used for sentiment analysis, including naive Bayes, logistic regression, multinomial logistic regression, ordinal regression models, proportional likelihood model, support vector machines, random forests, and eXtreme gradient boost. In addition, the section discusses the performance metrics and validation techniques used to evaluate the effectiveness of the models.

The Case Study section presents the data and methods used to conduct a practical analysis of sentiment in the retail industry. It describes the dataset utilized for the study, including its sources and characteristics. The section then outlines the specific methods and techniques employed to perform sentiment analysis on the data. Results and discussions of the findings are presented, providing insights into the effectiveness of the chosen models and their performance in classifying sentiment.

The final section summarizes the key findings and conclusions drawn from the research. It highlights the contributions made to sentiment analysis in the retail industry and discusses the implications of the results. Furthermore, this section suggests potential avenues for future research.

Chapter 2

Literature Review

2.1 Literature Review

Sentiment analysis, a crucial aspect of natural language processing, informs decision-making by deciphering public sentiment. It is applied across sectors including product analysis, healthcare, finance, and business, assisting in understanding customer sentiment, reputation management, and market trend prediction [Mowlaei et al. 2020, Kumar and Uma 2021, Ruffer et al. 2020, Park et al. 2020, Cortis and Davis 2021, Arora et al. 2021, Ahmad et al. 2019]. However, the process faces obstacles such as informal writing styles, sarcasm, irony, and language-specific nuances, complicating sentiment detection and classification [Subhashini et al. 2021, Wankhade et al. 2020].

In this study, it is proposed a hybrid approach combining SentiLex-PT – a comprehensive Portuguese sentiment lexicon [Pereira 2021, Ranchhod et al. 1999, Barreiro et al. 2015] – and diverse machine learning techniques. A sentiment lexicon associates each word with an opinion polarity or emotion, proving invaluable for sentiment polarity classification. SentiLex-PT, specifically designed for social judgment, facilitates sentiment extraction and interpretation from text. Additionally, machine learning techniques including naive Bayes, multinomial logistic regression, ordinal regression, support vector machines, random forests, and eXtreme gradient boost help classify sentiments into distinct categories.

Our approach leverages `SentiLex-PT` for sentiment extraction and machine learning techniques for sentiment classification, proving effective when analyzing Portuguese supermarket customer feedback. Document-level analysis provides an overall sentiment, while sentence-level analysis captures contrasting sentiments within the same document, offering a nuanced understanding. Phrase-level sentiment anal-

ysis enables a more detailed scrutiny by identifying opinion words at the phrase level [Thet et al. 2010]. The goal is to provide a comprehensive sentiment understanding by integrating these analysis levels.

Beyond text preprocessing, sentiment analysis also necessitates feature extraction, which converts raw text into a numerical format suitable for machine learning algorithms. This process incorporates methods like Bag-of-words, context capturing with N-grams, and sparse matrix management through feature selection and reduction. It also includes the application of evaluation metrics like precision, recall, F1-score, and ROC AUC, and hyperparameters tuning using methods like grid search and randomized search.

Machine learning models were carefully selected considering the sentiment analysis problem nature and our dataset characteristics. Multinomial logistic regression was used for nominal sentiment categorization, while ordinal logistic regression was employed for sentiments with inherent order. SVM were chosen due to their aptitude with high-dimensional data. Random forests, robust against overfitting and adept at managing large feature sets, were used to model complex feature interactions. Lastly, the eXtreme gradient boosting model, known for its regularization parameters, was employed to reduce overfitting and enhance performance, demonstrating superior predictive power, speed, and an advanced implementation of the gradient Tree-Boosting algorithm.

2.2 Machine Learning and Lexicon-Based Approaches

Supervised machine learning is a method where algorithms are trained on labeled data, i.e., a dataset comprising both the input and corresponding output values. This dataset essentially acts as an example for the algorithm to learn from. "Supervised" refers to the process where these known, correct input-output pairs guide the learning process.

The objective of these algorithms is to discover underlying patterns or relationships within the provided data, enabling the construction of a model that can accurately predict the output for new, unseen inputs. This is achieved through an iterative process of making predictions and receiving feedback. The feedback is an error signal indicating the discrepancy between the algorithm's predictions and the actual values.

Based on this feedback, the algorithm adjusts its internal parameters in a process commonly referred to as learning, aiming to minimize the deviation between its

predictions and the true values. This cycle of prediction, feedback, and adjustment continues until the algorithm achieves an acceptable level of performance, or until further learning ceases to significantly reduce the error.

Through this supervised learning process, the machine learning model can generalize from the training data and accurately predict when presented with new, similar data. However, while the algorithm can learn from labeled data and make precise predictions for unseen data, the operator (or machine learning practitioner) has the responsibility to evaluate and verify the model's performance and decide when the learning process should be halted.

Machine learning approaches handle sentiment classification as a standard text classification problem, employing syntactic and/or linguistic features. These approaches formulate categorization models linking the attributes of a given record to one of the class labels. They predict the class label for an instance of an unknown class, either attributing one label (hard categorization) or producing probabilistic label values (soft classification). Machine learning permits systems to acquire new capabilities without explicit programming, allowing sentiment analysis algorithms to understand contextual information, sarcasm, and misused words beyond mere definitions.

In [Hassonah et al. 2020] the authors used Twitter data for training, demonstrating their model's effectiveness by reducing the total feature count by up to 96% while outperforming most models. Their hybrid model underscored the potential of meticulously architected models with precision-tuned hyperparameters to surpass standalone models in performance [Chang et al. 2020]. While their hybrid model yielded impressive results, they recognized that opportunities for enhancing performance remain through model tweaking and training. By adopting such a hybrid approach in our study, the point is to advance sentiment analysis, amalgamating linguistic feature incorporation and sophisticated machine learning methods for more refined and precise sentiment classification.

In order to fill gaps between the current state of the literature and the problem this dissertation aims to answer, the article [Almeida et al. 2023], submitted for publication, was also used as support.

Chapter 3

Methodology

3.1 Models

In this section, it is described the machine learning classifiers utilized in our approach.

3.1.1 Naive Bayes

Naive Bayes is a simple yet effective machine learning algorithm for classification. The algorithm is based on the Bayes theorem and assumes that all features are independent of each other. Despite this oversimplification, naive Bayes classifiers work well in many real-world situations, including text classification and spam filtering [Rish 2001, Hand and Yu 2001, Mitchell 1997, Lewis 1998].

Bayes' theorem forms the core of this algorithm, stated as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.1}$$

Here, $P(A|B)$ is the posterior probability of class (A, target) given predictor (B, attributes), $P(B|A)$ is the likelihood which is the probability of predictor given class, $P(A)$ is the prior probability of class, and $P(B)$ is the prior probability of predictor.

The Naive Bayes model assumes that given a class variable y and dependent feature vector x_1 through x_n, the following condition holds:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{3.2}$$

Given the naivety assumption that all features are independent of each other, the conditional probability can be rewritten as follows:

$$P(y|x_1, ..., x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{3.3}$$

While constructing the classifier, the denominator $P(x_1, ..., x_n)$ doesn't actually matter, as it is a constant given the input. So, it can be rewritten as:

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.4}$$

So, it is chosen the class y that maximizes the $P(y) \prod_{i=1}^{n} P(x_i|y)$ equation, which leads to the final form:

$$y = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.5}$$

The naive Bayes model is easy to build, with no complicated iterative parameter estimation schemes that make it particularly useful for very large datasets. Besides, it can handle an extremely large number of features and is unaffected by irrelevant features, which makes it quite versatile in handling complex classification problems.

3.1.2 Logistic Regression

Logistic regression is a powerful and commonly used machine learning algorithm used for binary classification problems. It is a statistical model that utilizes a logistic function to model a binary dependent variable (see [Agresti 2010] for a more detailed explanation).

Unlike linear regression, which outputs continuous values, logistic regression transforms its output using the logistic sigmoid function to return a probability value, which can be mapped to two or more discrete classes.

Given an input vector x, the logistic regression model first calculates a linear combination of the input features, as follows:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{3.6}$$

where $\beta_0, \beta_1, ..., \beta_n$ are the parameters of the model, $x_1, ..., x_n$ are the feature values, and z is known as the log-odds or logit.

In the next step, the model maps the calculated log-odds to the probability of the positive class using the sigmoid function:

$$\hat{p} = \frac{1}{1 + e^{-z}} \tag{3.7}$$

The output $\hat{p}$ is the estimated probability of the positive class.

In order to train the model, is necessary to find the parameters β that minimize the cost function. This cost function is typically the log-loss function, which for two classes is defined as:

$$J(\beta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)] \tag{3.8}$$

where y_i is the actual class label and $\hat{p}_i$ is the predicted probability of the positive class for the i^{th} instance.

This is a convex cost function, so gradient descent (or any other optimization algorithm) is guaranteed to find the global minimum.

Once the model has been trained and our optimized β values are obtained, predictions can be made by applying these coefficients to the feature values, applying the sigmoid function to the result, and classifying instances with a probability greater than 0.5 as the positive class.

$$\widehat{y} = \begin{cases} 1 & \text{if } \widehat{p} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Logistic regression is widely used due to its efficiency and simplicity. Despite its name, it is primarily used for classification tasks rather than regression tasks. It works well for linearly separable classes and when the feature space is linearly influential to the log odds of the response variable. However, it may not be as effective with non-linear problems or problems where feature interactions are important, as it assumes independence among the features.

3.1.3 Multinomial Logistic Regression

Multinomial logistic regression is an extension of binary logistic regression used when the response variable has more than two unordered categories. It creates a distinct logistic regression model for each category against a reference category, each with its own set of intercepts and regression coefficients. The key assumption, independence of irrelevant alternatives, posits that the selection of one category does not relate to the selection of any other.

Given a categorical dependent variable Y with five levels (Very Negative, Negative, Neutral, Positive, and Very Positive) and m predictors $X_1, X_2, ..., X_m$, the MLR models are as follows:

$$y_1 = \log \frac{P(Y = \text{Very Negative}|X)}{P(Y = \text{Neutral}|X)} = \beta_{0,1} + \beta_{1,1} \cdot X_1 + \beta_{2,1} \cdot X_2 + \ldots + \beta_{m,1} \cdot X_m \quad (3.10)$$

$$y_2 = \log \frac{P(Y = \text{Negative}|X)}{P(Y = \text{Neutral}|X)} = \beta_{0,2} + \beta_{1,2} \cdot X_1 + \beta_{2,2} \cdot X_2 + \ldots + \beta_{m,2} \cdot X_m \quad (3.11)$$

$$y_3 = \log \frac{P(Y = \text{Positive}|X)}{P(Y = \text{Neutral}|X)} = \beta_{0,3} + \beta_{1,3} \cdot X_1 + \beta_{2,3} \cdot X_2 + \ldots + \beta_{m,3} \cdot X_m \quad (3.12)$$

$$y_4 = \log \frac{P(Y = \text{Very Positive}|X)}{P(Y = \text{Neutral}|X)} = \beta_{0,4} + \beta_{1,4} \cdot X_1 + \beta_{2,4} \cdot X_2 + \ldots + \beta_{m,4} \cdot X_m \quad (3.13)$$

Since $P(Y = \text{Very Negative}|X) + P(Y = \text{Negative}|X) + P(Y = \text{Neutral}|X) + P(Y = \text{Positive}|X) + P(Y = \text{Very Positive}|X) = 1$, we can derive all probabilities as:

$$P(Y = \text{Very Negative}|X) = \frac{\exp(y_1)}{1 + \exp(y_1) + \exp(y_2) + \exp(y_3) + \exp(y_4)} \quad (3.14)$$

$$P(Y = \text{Negative}|X) = \frac{\exp(y_2)}{1 + \exp(y_1) + \exp(y_2) + \exp(y_3) + \exp(y_4)} \quad (3.15)$$

$$P(Y = \text{Neutral}|X) = \frac{1}{1 + \exp(y_1) + \exp(y_2) + \exp(y_3) + \exp(y_4)} \quad (3.16)$$

$$P(Y = \text{Positive}|X) = \frac{\exp(y_3)}{1 + \exp(y_1) + \exp(y_2) + \exp(y_3) + \exp(y_4)} \quad (3.17)$$

$$P(Y = \text{Very Positive}|X) = \frac{\exp(y_4)}{1 + \exp(y_1) + \exp(y_2) + \exp(y_3) + \exp(y_4)} \quad (3.18)$$

These equations set the foundation for modeling the sentiment categories in our MLR framework. Following the derivation of the five-class sentiment model, we proceed with its fine-tuning and assessment. We adopt the maximum likelihood estimation method for fitting the multinomial logistic regression model. The significance of the fitted model is then evaluated using the likelihood ratio Test. This test enables us to compare the goodness of fit of two models by calculating the ratio of their likelihoods - the likelihood of the reduced model (with only an intercept) and the likelihood of the full model (with all predictors) [Hosmer et al. 2013].

When assessing the goodness of fit, we employ the Pseudo-R2 measures proposed by Cox & Snell, Nagelkerke, and McFadden. The interpretation of these Pseudo-R2 values isn't well-defined in the literature [Osborne 2017, Pituch et al. 2015]. However, they are instrumental when comparing competing models derived from the same dataset - the model with the highest Pseudo-R2 statistic is deemed to be the best fit. McFadden himself suggested that a pseudo-R2 between 0.2 and 0.4 is indicative of an excellent model fit [Hensher and Stopher 2021].

The significance of the model coefficients is examined with the Wald test, while the receiver operating characteristic curve helps evaluate the model's discriminative capacity. The area under the ROC curve ranges from 0 to 1, where an AUC of 0.5 suggests the model lacks discriminative ability; for $0.5 < \text{AUC} < 0.7$, the discriminative power is weak; for $0.7 \leq \text{AUC} < 0.8$, it's acceptable; for $0.8 \leq \text{AUC} < 0.9$, it's good; and for $\text{AUC} \geq 0.9$, the discriminative power is exceptional [33].

It's essential to note that, unlike other statistical procedures, multinomial logistic regression requires careful attention to sample size, especially in the presence of potential collinearity among the predictors. Care is needed with small sample sizes and highly correlated predictors, as these can lead to incorrect or unreliable inferences based on the fitted regression model [Ashqar et al. 2021]. As a general guideline, it's recommended to consider at least 100 cases for maximum likelihood estimation, including logistic regression. However, 500 cases are considered adequate in most applications, and it's suggested to have at least 10 cases per predictor [Long 1997].

3.1.4 Ordinal Regression

Ordinal logistic regression, also known as ordinal regression, is a type of regression analysis used when the dependent variable is ordinal, i.e., it has multiple ordered categories. Unlike nominal categories, ordinal categories have a specific order (say, for example, "very negative", "negative", "neutral", "positive", "very positive"), but the distance between the categories is not known [Agresti 2010, McCullagh 1980].

Before fitting ordinal regression models, we evaluated the association between the response variable and covariates using the Cochran–Mantel–Haenszel (row mean scores) statistic. This statistic explores the connection between the ordinal response variable and a particular covariate, adjusting for the effect of another covariate treated as a stratification variable. The ordinality of the response variable is incorporated by assigning scores to response categories, computing means, and inspecting location shifts of means across row levels or sub-populations (formed when covariate levels are cross-classified). Moreover, to account for ordinality in the face of uncertainly equally-spaced y-response categories, we assigned modified ridit scores.

3.1.5 Proportional Odds Model

The proportional odds model, also known as the cumulative logit model, is especially relevant when dealing with a grouped continuous response variable. Considere a c-point scale, let the response categories be denoted by $y1$, $y2$,...,yc and $X1$, $X2$,..., Xp be a set of explanatory variables or covariates. If we form cumulative probabilities from the proportions $Pr(Y = y_j) = p_j (j = 1, \ldots, c)$ based on the marginal totals of a sub-population, the probability of a response in category y_j or below can be represented as $Pr(Y \leq y_j) = p_1 + p_2 + \cdots + p_j$. As such, $Pr(Y \leq y_1) \leq Pr(Y \leq y_2) \leq \cdots \leq Pr(Y \leq y_c) = 1$ holds, reflecting the ordering in the response categories. The proportional odds model is built upon these cumulative probabilities. It defines a relationship between these cumulative probabilities and the covariates using the logistic function, as shown below:

$$\text{logit}\big(Pr(Y \leq y_j)\big) = \alpha_j - \beta \cdot X \tag{3.19}$$

Where $\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$ is the logit function, $Pr(Y \leq y_j)$ is the cumulative probability of the response being in category j or lower, α_j is the threshold parameter for category j, β is a vector of coefficients for the covariates X, $\beta \cdot X$ represents the dot product of β and X. Note that the minus sign in front of $\beta \cdot X$ indicates that as the covariates increase, the cumulative probability decreases, which maintains the ordering of the categories.

The proportional odds aspect of this model comes from the fact that the odds ratio between different categories is constant. Formally, for any two sets of covariates X and X', we have:

$$\frac{\text{odds}\big(Pr(Y \leq y_j|X)\big)}{\text{odds}\big(Pr(Y \leq y_j|X')\big)} = \frac{\exp(\beta' X)}{\exp(\beta' X')} = \exp(\beta'(X - X')) \tag{3.20}$$

This odds ratio is constant for all categories j, meaning that a unit increase in any covariate multiplies the odds of being in a lower category by a constant factor, regardless of the current category. This is a strong assumption and may not hold in all datasets, so it's important to check this assumption when using the proportional odds model.

Multinomial logistic vs. Ordinal regression

If the categories have a clear order (like the example given), ordinal logistic regression might be more suitable since it takes this order into account. On the other hand, multinomial logistic regression treats each category as a separate class without considering any order or hierarchy. Hence, ordinal regression might capture the nature of sentiment analysis better than multinomial logistic regression.

The ordinal logistic regression model can give meaningful results about shifts from one category to another, considering the order. In comparison, multinomial logistic regression will treat shifts from "very negative" to "negative" the same as from "very negative" to "very positive". This can be less intuitive in a sentiment analysis context, where we typically think of sentiment as a continuum from negative to positive.

The performance of both models can depend on the specifics of the data. Multinomial logistic regression might perform better when the assumptions of ordinal logistic regression do not hold, or when the order of categories is not very meaningful for predicting the outcome. However, when the ordinal nature of the categories is important, ordinal regression can outperform multinomial regression.

Ordinal logistic regression requires the assumption of proportional odds, meaning that the odds are the same regardless of the cut-off point chosen to distinguish between 'lower' and 'higher' ratings. Checking and handling violations of this assumption can add to the complexity of using ordinal regression. In contrast, multinomial logistic regression doesn't make this assumption, which can make it easier to use.

3.1.6 Support Vector Machines

Support Vector Machines is a supervised machine learning algorithm mainly used for classification problems and also for regression [Vapnik 1995, Cortes and Vapnik 1995]. When it comes to sentiment analysis, SVM can be a powerful tool.

(1) SVMs are particularly good at handling high-dimensional data. In the context of text data for sentiment analysis, after pre-processing, the dataset will likely have a very high number of features (given by the size of the vocabulary).

(2) Text data is typically very sparse - each document only contains a small fraction of the total vocabulary. SVMs work well with sparse data and have been shown to outperform other algorithms in such situations.

(3) While SVMs are fundamentally binary classifiers, they can be extended to handle multi-class problems like our 5-category sentiment analysis. Common strategies are "one-vs-one" and "one-vs-all" approaches.

(4) SVMs are robust against overfitting, especially in high-dimensional space. This is particularly useful for text data where the number of features can be very high.

The primary idea behind SVM is to find a hyperplane that best separates the data into its respective classes. In a 2-dimensional space, this hyperplane is simply a line. For data that is linearly separable (i.e., it is possible to draw a straight line to separate different classes), an SVM will find the line that maximizes the margin between the closest points (support vectors) of each class. This line is known as the decision boundary, and the area defined by the margin is called the "street". For non-linearly separable data, SVM uses a technique known as the kernel trick. The kernel trick involves transforming the data into a higher-dimensional space where it can be separated by a hyperplane. Once the hyperplane is found in this higher-dimensional space, we can then transform it back to the original space, resulting in a non-linear decision boundary.

In practice, data might be noisy and classes might overlap, making it impossible to find a hyperplane that perfectly separates the classes. To handle such situations, SVM introduces the concept of a "soft margin" which allows some points to be on the wrong side of the margin or even the wrong side of the decision boundary. The degree to which this is allowed is controlled by a parameter C (regularization parameter), which is tuned to find the best balance between maximizing the margin and minimizing misclassification.

Although SVMs are mostly known for classification, they can also be used for regression tasks (Support Vector Regression or SVR). The principle is similar, but instead of trying to find the largest possible margin between two classes while minimizing misclassification, SVR tries to fit as many instances as possible within the

margin while minimizing the number of instances outside the margin. The width of the margin is controlled by a parameter epsilon.

3.1.7 Random Forests

The Random Forest classifier is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees [Breiman 2001, Breiman and Cutler 2004].

The Random Forest algorithm involves the following steps:

1. The procedure starts by picking n random samples from the dataset with replacement (known as bootstrap samples), where n is the total number of observations in the dataset.

2. For each bootstrap sample, a decision tree is grown. At each node:

   (a) m variables (features) are selected at random out of all p variables. By default, m is equal to the square root of p.

   (b) The best split on these m is used to split the node. The objective of the split is to maximize the reduction in variance (for regression trees) or Gini impurity (for classification trees). The node is split into left and right child nodes.

   The splitting process is mathematically represented as:

$$\Delta I(S, A) = I(S) - \sum_{\nu \in \text{Values}(A)} \frac{|S_\nu|}{|S|} I(S_\nu)$$

   where $I(S)$ is the impurity measure of the original set, Values(A) is the set of all possible values for attribute A, S_ν is the subset of S for which attribute A has value ν, $\frac{|S_\nu|}{|S|}$ is the proportion of the examples that have value ν for attribute A. For each attribute, the potential information gain is calculated and the attribute with the highest information gain is chosen for the split.

3. For prediction with a new object, it's passed down all of the trees in the forest. Each tree gives a classification, and the tree votes for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and outputs that as its prediction.

A crucial parameter of the Random Forest is the number of trees to generate (typically denoted as `n_estimators` in software implementations). A larger number of trees reduce the likelihood of overfitting but increases the computational complexity.

Random Forests are robust to outliers, scalable, and able to naturally model non-linear decision boundaries thanks to their hierarchical structure. Random Forests can also be a good choice for sentiment analysis, because can handle high-dimensional datasets well, which can be very useful in sentiment analysis when using bag-of-words or TF-IDF based feature vectors, resulting in a large number of features and automatically give an estimate of what variables are important in the classification. Also are capable of modeling complex, non-linear decision boundaries and do not require input features to be scaled (i.e., to have the same range), unlike methods such as SVMs or logistic regression.

However, RF, being a black-box model, are not as interpretable as models like logistic regression. It can be harder to understand why the model is making certain predictions. As SVM, RF can be more computationally intensive and require more memory, especially as the number of trees increases or the depth of the trees increases. When dealing with sparse data (like text data typically is), RF may not perform as well as linear models or SVMs.

3.1.8 eXtreme Gradient Boost

The eXtreme Gradient Boost was developed in the paper [Chen and Guestrin 2016], is an enhancement of the gradient tree boosting algorithm, using regularization to mitigate the risk of overfitting.

XGBoost, as with other boosting algorithms, operates by aggregating the performance of weak learners to create a single strong learner. The weak learners are, in this case, decision trees, which individually might not produce the best predictions but collectively deliver a high performance. These decision trees are designed to minimize a loss function, with each subsequent tree working on reducing the errors left by the preceding trees.

The essential element of the XGBoost algorithm is its regularization component. Regularization is a technique used in machine learning to avoid overfitting, which occurs when a model learns the training data too well, including its noise and outliers, hence performing poorly on unseen data. By adding regularization terms into the loss function, XGBoost encourages the model to become simpler or smoother, meaning it will prefer simpler models over complex ones, thereby reducing overfitting.

The XGBoost classifier operates as a meta-classifier that amalgamates weak learners to construct a robust learner. For a given training dataset X_i, with corresponding labels Y_i, the XGBoost classifier leverages individual classifiers to predict the outcome Z_i. The prediction equation is represented as:

$$Z_i = \sum_{n=1}^{N} f_n(X_i)$$

where function f_n symbolizes de n^{th} tree, housing scores on its leaves. The score of each tree is computed through the following function:

$$L(n) = \sum_{k=1}^{k} l\left(Y_i, Z_i^{(n-1)} + f_n(X_i)\right) + \Omega f(n).$$

Here, l denoes the loss function, $Z_i^{(n)}$ indicates the prediction for sample X_i at the n^{th} iteration, and Ω signifies the regularization term which inhibits the score leaves from acquiring large values. The $f_n(X_i)$ function that minimizes the aforementioned equation is integrated into the tree function, generating the final classification tree.

The efficiency of the XGBoost algorithm lies in its computation speed and performance. It has been empirically shown to be faster and more predictive than traditional Gradient Tree Boosting algorithms (Dhaliwal, Nahid, and Abbas, 2018). This is primarily due to its unique system design and the implementation of a column block structure, which enables parallelization during model construction. Additionally, it can handle sparse data and missing values, making it more versatile and robust.

To sum up, the XGBoost classifier employs gradient boosting techniques while incorporating regularization, which makes it highly effective for various complex predictive tasks. This robust learner, with its solid theoretical foundation and proven effectiveness, continues to be a significant tool in the machine learning field.

Random forests vs. XGBoost

While both XGBoost and Random forests are ensemble machine learning algorithms that use decision trees as their base learners, there are some fundamental differences between the two in terms of their learning process, bias-variance tradeoff, and their handling of feature importance.

Random Forests build each tree independently while XGBoost builds one tree at a time, where each new tree helps to correct errors made by the previously trained

tree. This sequential nature of boosting algorithms can make them more sensitive to noise and outliers, but can also make them more accurate if tuned correctly.

Random Forests tend to reduce variance by averaging the results from a number of independently built decision trees, which makes them less prone to overfitting. On the other hand, XGBoost tackles both bias and variance by building a strong predictive model in a step-wise, additive, and sequential manner. This approach tends to make XGBoost more accurate than Random Forests, given that the parameters are properly tuned.

Random Forests consider a random subset of features for splitting nodes while building the trees, which leads to a diverse set of models and helps to reduce overfitting. XGBoost, in contrast, uses all features for splitting nodes while controlling complexity using its regularisation parameters.

Both Random Forests and XGBoost provide measures of feature importance, but the methods differ. In Random Forests, importance is based on the total decrease in node impurities from splitting on the variable, averaged over all trees. XGBoost, on the other hand, measures importance by the average gain of the feature when it is used in trees.

In terms of raw performance, XGBoost often has an edge over Random Forests. Because of its gradient boosting framework, XGBoost can often provide a better predictive accuracy than Random Forests. However, XGBoost models may take longer to train and tune due to their sequential nature.

3.2 Performance Metrics and Validation Techniques

Once the model is fit, or train, adequately on a dataset, evaluating its predictive performance and reliability is critical. This assessment involves several metrics and techniques, such as cross-validation or a simple train-test split, as explain in this subsection (see [Hastie et al. 2009] for a more detailed explanation).

Precision and sensitivity, also called recall, two commonly used metrics in retrieval tasks, are primarily focused on positive predictions. Precision is the ratio of correctly predicted positive observations to the total predicted positives, and sensitivity is the ratio of correctly predicted positive observations to all observations in the actual positive class. They are mathematically defined as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.21}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.22}$$

Accuracy, another significant metric, is the proportion of correct predictions made by the model, calculated as:

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total no. of predictions}} \tag{3.23}$$

For binary classification tasks, accuracy is calculated specifically as:

$$\text{Accuracy (Binary)} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.24}$$

where TN, TP, FP, FN refer to True Negative, True Positive, False Positive, and False Negative predictions, respectively. This is summarized in the confusion matrix table:

	Predictive Positive	Predictive Negative
Actual Positive Instances	True Positive (TP)	False Negative (FN)
Actual Negative Instances	False Positive (FP)	True Negative (TN)

Table 3.1: Confusion matrix calculation method

In cases of class imbalance, the F1 Score is employed. It combines precision and recall into a single metric and is defined as

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

F1 score is equal to 1 only when precision and recall are both 1. F1 score becomes high only when both precision and recall are high. F1 score is the harmonic mean of precision and recall and is a better measure than accuracy, specifically when considering unbalanced datasets.

The model can be assessed using the k-Fold Cross Validation resampling technique. The dataset is divided into k subsets, where k could be in the range of 2 to 20, for example.

Finally, learning curves, representing ”risk or cost / score vs size” for both the training and testing data, can be deployed. These curves assist in determining the volume of data required for optimal training in future iterations.

Chapter 4

Case Study

4.1 Data and Methods

In this section, it will be described the data and methods used in the project.

4.1.1 Data

For this project, a dataset comprising customer support related data was considered, which included the message sent by the customer, the message identification number and type, and the associated sentiment. It is important to note that this classification, divided into five classes and ranging from very negative to very positive sentiment, was performed by a former consultancy of the company.

The initial dataset consisted of 58,172 records, out of which 524 were identified as duplicate cases, that is, the same record repeated more than once. Additionally, there were 11,996 records that were not classified, and 654 cases with the exact same message. After addressing these issues, the final dataset consisted of 44,998 cases. Out of the approximately 45k cases, around 78% were categorized as service requests, 14% as information requests, 8% as complaints, and the remaining cases were classified as compliments and suggestions.

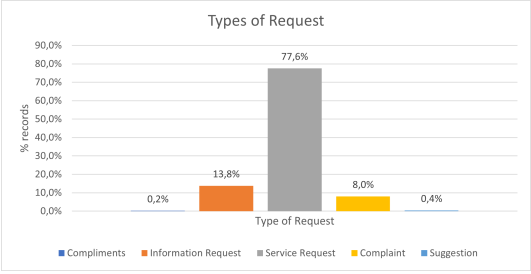


Figure 4.1: Types of request

Regarding the number of cases by sentiment, approximately 79% of the cases are classified as neutral. This is to be expected since the majority of cases are categorized as service requests and information requests and, typically, when a customer writes a message of this nature, their satisfaction tends to be neutral. Additionally, 20% of the records contain negative messages, while the remaining cases are divided among very negative, positive, and very positive sentiments. It is worth noting that the number of cases with a very positive sentiment is quite low, indicating the customers' inclination to write positive comments is limited.

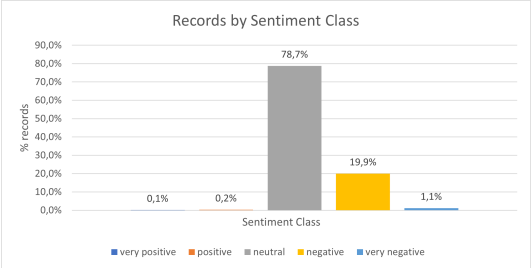


Figure 4.2: Records by sentiment class

Furthermore, by analyzing the Table 4.1, it becomes evident that, as expected, more positive sentiments are associated with messages categorized as compliments, service requests, and information requests, while more negative sentiments are associated with complaints and service requests as well. Additionally, upon examining the Figure 4.2, it is notable that approximately 10% of the cases classified as positive are categorized as complaints. A brief analysis of some of these cases reveals that this misclassification can occur due to the use of irony by the customer and the inclusion of positive words in their negative message, such as *bom dia*, *agradeço* and *cumprimentos*.

	Compliment	IR	SR	Complaint	Suggestion	Marginal Total
Very Positive	33.3%	12.5%	20.8%	0.0%	0.0%	24
Positive	9.8%	29.4%	50.0%	9.8%	1.0%	102
Neutral	0.2%	14.9%	78.4%	6.1%	0.4%	35412
Negative	0.0%	9.2%	75.5%	14.8%	0.4%	8964
Very Negative	0.0%	11.5%	64.5%	23.4%	0.6%	496

Table 4.1: Percentage of request types by sentiment class

4.1.2 Methods

Cleaning Process

With the data ready to be processed, the first step involves cleaning the messages, which entails removing irrelevant text and characters. This process is essential in preparing the data for subsequent analysis. Cleaning the messages aims to eliminate unwanted elements such as punctuation, numbers, special characters, and stop words (commonly used words such as *e*, *o* and *para*) that do not contribute significantly to sentiment analysis. Properly cleaning the messages is crucial for improving data quality and, consequently, obtaining more accurate and reliable results in sentiment analysis.

Here is an example of all the message cleaning processes used and tested in the project.

"Bom dia,\nGostaria de saber quando terão cadeiras, novamente, disponiveis.\nMuito obrigada!\n https://www.continente.pt/produto/cadeira-echair-confort-kasa-284190.htmml\n\nMelhores cumprimentos,\n\nMaria Silva, mariasilva@gmail.com \n\nDe: Continente <ajuda@continente.pt>\nEnviado: 25 de setembro de 2021 12:06 \nPara: Maria silva >mariasilva@gmail.com>\nAssunto: Confirmação de Encomenda\n\n"

↓

When importing the data, it is important to note that all spaces are automatically replaced with \n. Therefore, the initial process of cleaning plays a crucial role in restoring the spaces in the original messages. This step aims to correct the unwanted substitution of spaces with \n and ensure that the messages are returned to their original form, preserving the proper structure and readability.

"Bom dia, Gostaria de saber quando terão estas cadeiras, novamente, disponiveis. Muito obrigada! https://www.continente.pt/produto/cadeira-echair-confort-kasa-284190.html Melhores cumprimentos, Maria Silva, mariasilva@gmail.com —- De: Continente <ajuda@continente.pt> Enviado: 25 de setembro de 2021 12:06 Para: Maria Silva <mariasilva@gmail.com> Assunto: Confirmação de Encomenda"

↓

In the second cleaning step, all links and characters are removed except for '.', '!', and '@'. The period and '@' cannot be removed in this stage as they will be needed to remove email addresses in a subsequent step. As for the exclamation point, it was chosen not to remove it, as it can be relevant for identifying the sentiment in the message, as it is often used to express admiration or indignation. Furthermore, all uppercase letters are converted to lowercase. This is done to standardize the text and avoid discrepancies in the subsequent analysis. Converting all letters to lowercase, ensures that words are treated consistently, regardless of the capitalization used by the message sender.

"bom dia gostaria de saber quando terão estas cadeiras novamente disponiveis. muito obrigada! melhores cumprimentos Maria Silva mariasilva@gmail.com de continente enviado de setembro de para maria silva assunto confirmação de encomenda"

↓

It was noticed that many customer messages are responses to messages from the retailer *Continente* and therefore, after the customer's message, there is *Continente's* message, which in this case, is not relevant to understanding the customer's sentiment. Therefore, in the third cleaning step, it was removed the message sent by the retailer customer support. This ensures that the focus is solely on the customer's message and allows for a more accurate analysis of the customer's sentiment.

"bom dia gostaria de saber quando terão estas cadeiras novamente disponiveis. muito obrigada! melhores cumprimentos maria silva mariasilva@gmail.com"

↓

The fourth cleaning step involves removing email addresses since they do not provide relevant information for identifying the sentiment present in the customer's message. By removing email addresses, it's eliminated any potential noise or distractions from the analysis process and focus solely on the textual content that contributes to understanding the customer's sentiment.

”bom dia gostaria de saber quando terão estas cadeiras novamente disponiveis. muito obrigada! melhores cumprimentos maria silva”

↓

The fifth cleaning step is responsible for removing all accents and, finally, the periods. By removing accents, consistency is ensured in the text, and potential issues related to different accented characters are avoided.

”bom dia gostaria de saber quando terao estas cadeiras novamente disponiveis muito obrigada! melhores cumprimentos maria silva”

↓

In the sixth cleaning step, greetings such as *bom dia*, *boa tarde*, and *boa noite* are removed. It was chosen to remove these greetings as they are not indicative of a specific sentiment. It was observed that customers commonly include greetings in their messages, regardless of whether they are writing a positive or negative comment. This is also influenced by the common structure of emails, where a greeting is typically included before any type of message. By removing these greetings, the focus is directed towards the content that directly expresses the sentiment of the customer's message.

”gostaria de saber quando terao estas cadeiras novamente disponiveis muito obrigada! melhores cumprimentos maria silva”

↓

In the seventh cleaning step, stop words are removed. Stop words are words that commonly occur in a language and are responsible for connecting the text, but they typically do not carry significant meaning for sentiment analysis. These words often include determiners and pronouns. However, it is important to note that it was excluded the word *não* from the list of stop words, as it plays a crucial role in understanding the negative sentiment of a sentence. By removing stop words, unnecessary noise is eliminated from the text, allowing for a focus on the more meaningful words that contribute to sentiment identification.

”gostaria saber terao cadeiras novamente disponiveis obrigada! melhores cumprimentos maria silva”

↓

The eighth cleaning process concerns stemming, which is a language processing technique that reduces words to their base or root form. By reducing words to their stems, stemming helps to consolidate variations of a word into a single representation. This can be particularly useful in sentiment analysis, where the focus is on the underlying meaning of words rather than their specific forms.

”gost sab ter cade nov disponi obrigada! melhor cumpr mar silv”

Finally, it was noticed that some messages received by the Customer Service team come through the ”Portal da Queixa”, an online forum where customers can leave their messages. These messages are accompanied by an automated note from the *Portal da Queixa*, which is not relevant to the project. Therefore, in this last cleaning step, this note was removed from the messages. Eliminating this irrelevant information, ensures that the remaining text is solely focused on the customer's message, allowing for a more accurate sentiment analysis.

”O valor da encomenda nº 000000000 (05/07/2021) foi debitado duas vezes no meu cartão crédito. Fiz o pagamento através do MB Way. Peço a resolução desta situação com a maior brevidade. ———————————————————————————————- Nova reclamação recebida Olá, O Portal da Queixa rececionou uma reclamação por parte de um utilizador dirigida à marca Continente e que contém dados pessoais do(a) reclamante que recolhemos nos termos da nossa Política de Privacidade e Proteção de Dados (*https* : *//portaldaqueixa.com/*) ———————————————————————————————- Continente - Valor de encomenda online debitado duas vezes Reclamação 00000000 em 2021-07-07 18:44:52

↓

”valor encomenda debitado duas vezes cartao credito fiz pagamento atraves mb way peco resolucao desta situacao maior brevidade”

Feature extraction

After the messages have been cleaned and prepared for further processing, the next step is to extract relevant features from them that can be used to represent and characterize the data. By extracting relevant features, a compact and representative representation of the data is created, which can be used as input for various analysis tasks. These features serve as a condensed representation that captures the salient information, facilitating the uncovering of patterns, relationships, or trends within the data.

Number of words

The first feature to be extracted is the number of words in each message. This feature is an important aspect in sentiment analysis as it provides insights into the length and complexity of the text. The number of words in a message can be indicative of several factors that influence sentiment. The number of words can correlate with the intensity of sentiment. Longer messages often have a higher likelihood of containing emotional expressions, emphasizing specific points, or providing elaborate arguments. These linguistic cues can contribute to the overall sentiment expressed in the message.

Furthermore, the number of words can also reflect the level of engagement or investment of the author. Longer messages may indicate a higher degree of interest, passion, or dissatisfaction, which can significantly impact the sentiment passed on.

Number of letters

The number of letters in each message was also extracted for the same purpose. This is, once again, a feature that can provide insights into the expressed emotional intensity. Longer messages with a higher number of letters may indicate a greater level of detail, complex emotions, or a more intense expression of sentiment. On the other hand, shorter messages with a lower number of letters may indicate a more concise emotion or a more direct response. Therefore, the count of the number of letters can be useful in capturing nuances and variations in the sentiment expressed in the messages.

Number of vowels

As a relevant feature, the number of vowels in each message was also considered. Vowels play a significant role in conveying emotions and emphasizing words. By

counting the number of vowels in a message, insights can be gained into vocal intensity and potential emotional emphasis. Messages with a higher number of vowels may indicate a more passionate or intense expression, while messages with a lower number of vowels may suggest a more neutral or restrained emotional tone. Thus, considering vowel count can provide valuable information for understanding the emotional content and intensity present in the text.

Lexical diversity

A lexical diversity was also considered as a feature for this project. It represents the proportion of unique words compared to the total number of words in a message. A high percentage of unique words indicates a greater lexical variety and vocabulary diversity in the message. This can be indicative of a more elaborate expression, rich in details, and with a wide range of words to convey the sentiment. On the other hand, a low percentage of unique words suggests a higher repetition of words and a more limited vocabulary, which may reflect a simplified sentiment or a less complex message. Therefore, analyzing the percentage of unique words can provide insights into the lexical richness and complexity of emotional expression in the analyzed messages. Additionally, it is worth noting that a lower percentage of unique words can indicate a more negative sentiment, as dissatisfied customers tend to repeat the cause of their dissatisfaction.

For the following group of features, a pre-existing dictionary called `SentiLex_flex_PT02` was utilized. This dictionary is a widely used lexicon specifically designed for sentiment analysis in portuguese. It consists of a comprehensive collection of words with their associated sentiment scores. Each word in the lexicon is annotated with a polarity label indicating whether it is positive, negative, or neutral in terms of sentiment. The lexicon serves as a valuable resource for sentiment analysis, allowing us to quantify and analyze the sentiment expressed in the textual data.

The use of the `SentiLex_flex_PT02` enhances the interpretability and consistency of sentiment analysis results, as it is based on a well-established lexicon with reliable sentiment annotations. This allows for the identification of patterns, trends, and insights related to the sentiment expressed by customers, further enriching the analysis and providing valuable information for decision-making processes.

Taking this into account, based on this lexicon, features were extracted and named as sentiment score, positive score, negative score, and number of adjectives.

The first one is derived from the difference between the number of positive words and the number of negative words in the message. PS and NS represent the counts of positive and negative words, respectively, present in the text. Lastly, the NA feature captures the count of adjectives in the customer's message. These features provide valuable insights into the sentiment expressed by customers, allowing for a more detailed analysis of the emotional tone and linguistic characteristics of the text.

Text to number

Converting text to numbers is necessary because most machine learning algorithms and models operate on numerical data. Text, on the other hand, is a form of unstructured data composed of words, phrases, and paragraphs.

By converting text into numbers, textual data can be represented in a format that can be easily processed and analyzed by machine learning algorithms. This conversion allows us to extract meaningful features from the text and perform various natural language processing tasks, such as sentiment analysis.

In the process of converting words to numbers, the goal is to transform text into a numerical representation that can be understood by machine learning algorithms. For that, it was used the `Tokenizer` class from `Keras` to perform this transformation. The `Tokenizer` analyzes the text, breaks it down into individual words, and assigns a unique number to each word. It then creates a binary representation for each text, where each position in the matrix represents the presence or absence of a specific word in the text.

This binary representation is useful because it allows the model to know which words are present in a given text and which ones are not. This information can be used to learn patterns and relationships between words and, consequently, perform tasks such as text classification.

In the end, the result is a matrix where words are represented by binary numbers, enabling the model to process and understand the texts in a format suitable for sentiment analysis

Here is an example of converting the sentence *gosto de ir ao continente* into a binary vector.

continente	entrega	ir	muito	...	de	gosto	ao
1	0	1	0	...	1	1	1

Table 4.2: Conversion of the sentence *gosto de ir ao continente* into a binary vector

So, in the final database, the column corresponding to the customer's message is replaced by the vector $[1, 0, 1, 0, ..., 1, 1, 1]$.

Class balancing techniques

Class balancing techniques are employed in machine learning and data analysis to tackle the issue of imbalanced datasets, where one class is significantly over-represented compared to others. Imbalanced datasets can introduce challenges during model training and evaluation, as the model may exhibit a bias towards the majority class, resulting in subpar performance on the minority class(es). As evident from the data distribution in this project, the positive and neutral classes are highly imbalanced relative to the others. Therefore, it is pertinent to apply and evaluate several class balancing techniques, including:

Oversampling

This technique involves increasing the number of samples in the minority class by replicating existing samples or generating synthetic samples. This helps in balancing the class distribution and providing the model with more representative examples of the minority class.

Undersampling

This technique aims to reduce the number of samples in the majority class to match the size of the minority class. It involves randomly selecting a subset of samples from the majority class, ensuring a more balanced dataset.

Hybrid approaches

These techniques combine oversampling and undersampling to achieve a balanced dataset. They involve oversampling the minority class and undersampling the majority class simultaneously, striking a balance between the two.

Class weighting

This technique assigns higher weights to the minority class during model training, thereby emphasizing its importance and mitigating the impact of class imbalance.

Wordclouds

Wordclouds can be a valuable tool for sentiment analysis as they provide a visual representation of the most common words used in messages with a specific sentiment. Analyzing the wordcloud, give us insights into the prevalent themes and emotions associated with a particular sentiment. Wordclouds highlight the frequency of words by representing them in varying sizes, with larger words indicating higher occurrence. This allows us to quickly identify the prominent terms that contribute to a specific sentiment.

In the specific field of retail, wordclouds enable us to identify which products and services from a brand are most satisfying to customers and, conversely, which ones may cause dissatisfaction. By visually representing the most common words associated with a sentiment, wordclouds provide a clear indication of customers' preferences and opinions.

By analyzing these wordclouds, retail businesses can easily identify their strengths and areas for improvement. This information is invaluable for making data-driven decisions to enhance customer satisfaction and overall business performance. It allows businesses to focus on optimizing the aspects that customers appreciate the most while addressing any pain points that negatively impact their experience.

In Appendix A two examples of wordclouds generated for each sentiment class are presented.

N-grams

N-gram analysis is a valuable technique in sentiment analysis for the retail industry as it provides deeper insights into the context and relationships between words in customer feedback. N-grams refer to contiguous sequences of n words, where "n" represents the number of words in the sequence.

By analyzing n-grams in customer feedback, retail businesses can uncover meaningful patterns and associations that contribute to specific sentiments. This analysis helps in understanding the key phrases, expressions, and combinations of words that are commonly used by customers when expressing their sentiments.

N-gram analysis also enables businesses to identify sentiment-specific language patterns that may be unique to their industry or brand. By analyzing the sentiments associated with specific n-grams, retailers can gain insights into customer preferences, pain points, and sentiment shifts over time. This information can guide decision-making processes, inform product development strategies, and help tailor marketing campaigns to align with customer sentiments.

In Appendix B two examples of n-grams, specifically bi-grams e tri-grams, generated for each sentiment class are presented.

4.2 Results and Discussions

For this project, the data was tested on various models, including proposed models and pre-trained models. The purpose was to evaluate their performance in sentiment analysis and determine the most suitable model for the task. Through rigorous testing and analysis, valuable insights were gained into the strengths and weaknesses of each model. These findings will contribute to the development of an effective sentiment analysis system for the customer support team of the Portuguese retailer.

Proposed models

- Naive Bayes
- Random forests
- Multinomial logistic
- XGBoost
- Ordinal regression
- Support vector machines [1]

Pre-trained models

- TextBlob
- Vader
- Twitter Roberta Base Sentiment Latest

For the initial analysis, the three models, NB, RF and ML, were tested. In this preliminary stage, a total of 40,155 features were included in the models, representing the total number of words contained in the data.

Upon analyzing the initial results, as illustrated in Figure 4.6, it is noticeable that all three models exhibit poor predictive performance in the ”very positive”

[1] Due to the requirements of the model and the data, it was not possible to obtain results for the SVM models.

class. This can be attributed to the limited amount of data available in this class. Therefore, a decision was made to combine the two classes related to positive cases, namely, merging the data from the positive class with the data from the very positive class. On a business perspective, no different action would be tacken to differentiate positive from very positive class. As a result, only 4 classes remained: very negative, negative, neutral and positive.

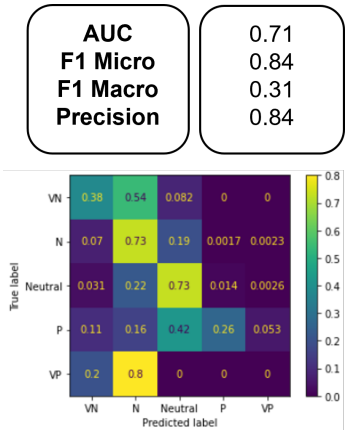


Figure 4.3: Initial results - NB

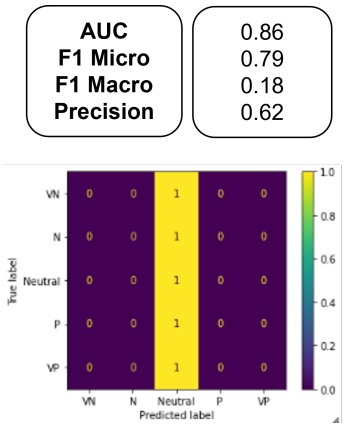


Figure 4.4: Initial results - RF

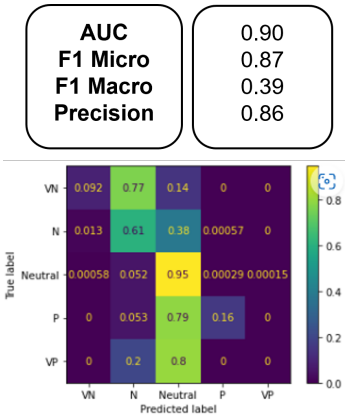


Figure 4.5: Initial results - ML

Figure 4.6: Initial results by model

After completing the process of merging the two classes with positive data, several tests were conducted considering different data treatment methods, such as class balancing techniques, as well as the application of two data cleaning processes: stop words removal and stemming.

#	Join Positives	Oversample	SMOTE	Undersample	Remove SW	Stemming
1	x	x				
2	x	x			x	x
3	x	x			x	
4	x			x	x	x
5	x				x	
6	x				x	x
7	x		x		x	
8	x		x		x	x

Table 4.3: Benchmarking Process

Upon completing the benchmarking process, it can be observed in Figure 4.10 that the best results for each model were obtained when applying the selected methods outlined in line three of the table 4.3. Specifically, when the minority class

experienced data resampling and when the stopword removal process was applied.

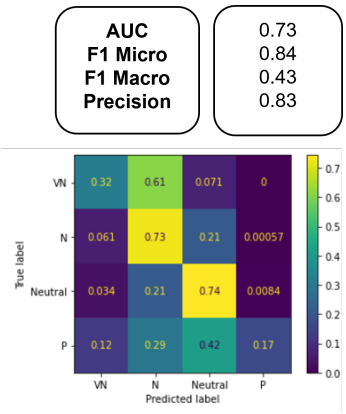


Figure 4.7: Second results - NB

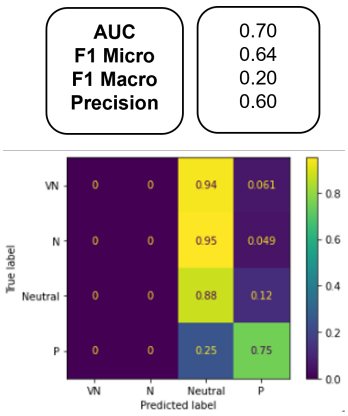


Figure 4.8: Second results - RF

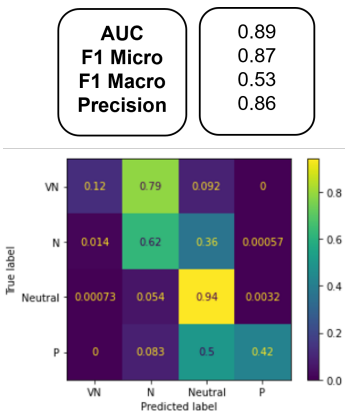


Figure 4.9: Second results - ML

Figure 4.10: Second results by model

Given the results, it was decided to choose the multinomial logistic regression model among these three models for further analysis, as it has higher AUC and precision values. This model also demonstrates better predictive capability in the positive and neutral classes, while still performing reasonably well in predicting negative data.

After completing this initial analysis and selecting the model to be used in the subsequent analyses, a second analysis was conducted where three additional features, sentiment score, positive score and negative score, were added to the existing 40155 features.

The results showed in Figure 4.11 indicate that the addition of these three features has improved the model's ability to predict negative, neutral, and positive data, resulting in an increased AUC value. Taking this into consideration, these features were included in the subsequent analyses.

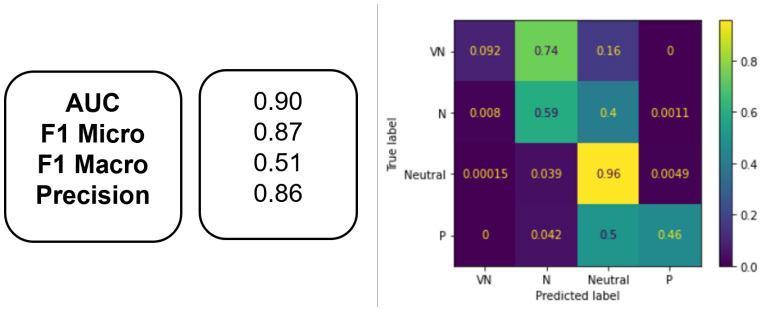


Figure 4.11: ML with the features SS, PS and NS

Afterward, the number of adjectives feature was also included as a variable, and

the results, that can be observed in Figure 4.12, showed an improvement in the AUC score. Therefore, this feature was incorporated into the selected set of features for the model.

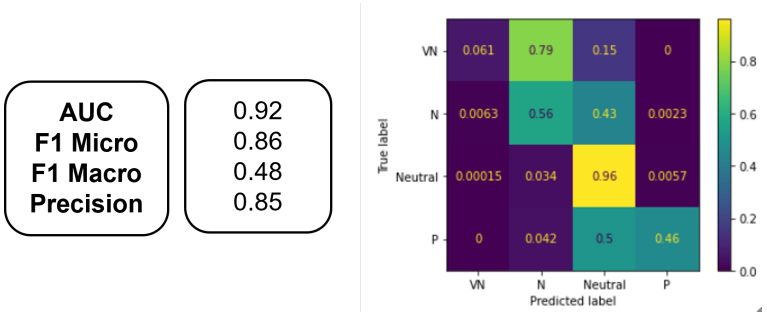


Figure 4.12: ML with the NA feature

In a subsequent analysis, the excessive number of features relative to the available training data, approximately 35,000 samples, was called into question. It was known that the large number of features could introduce noise to the model because, since the features correspond to words present in the customer's messages, it is possible that words appearing infrequently or even only once are being considered as features, which may not provide valuable information to the model. Therefore, an analysis was conducted to evaluate the model's performance when removing less frequent words from the features.

Upon evaluating the results of Figure 4.13, it is evident that removing less frequent words from the features indeed improves the model's performance. The model was able to increase its predictive capacity in the positive class by 8p.p. This finding highlights the significance of feature selection and demonstrates the impact of excluding less informative words from the analysis.

As the testing phase of the multinomial logistic model neared its conclusion, it was time to evaluate the class balanced parameter. This parameter allows the model to automatically adjust the weights of the classes during training to account for class imbalance in the dataset. By giving more weight to the minority class, it aims to address the challenges posed by imbalanced data. As the "class balanced" parameter already addresses the challenges caused by imbalanced data, it was compared to the model's performance with and without data oversampling.

The results presented in Figure 4.14 indicated that the class balanced parameter, regardless of whether oversampling was performed, contributed to improved performance in handling class imbalance. Specifically, in the negative class, the model's predictive capability increased significantly from 59% to 74% in both cases, and in the very negative class, it increased from 1% to 22% in both cases as well. However,

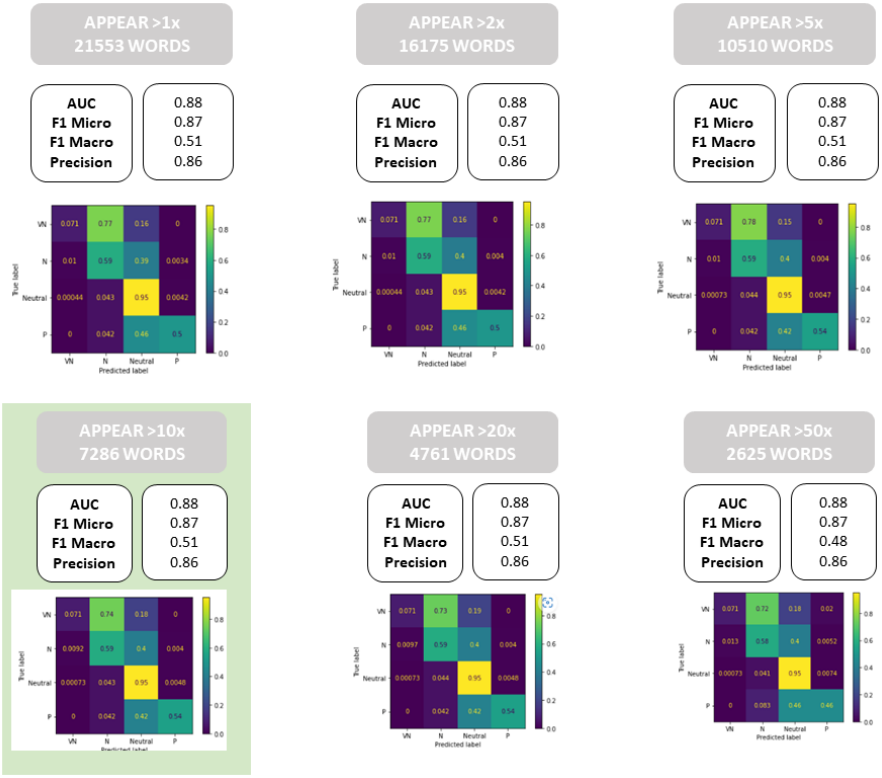


Figure 4.13: Removal of words from the features

despite observing significant improvements in AUC score and precision in both situations, it was decided not to apply both techniques simultaneously due to a slight decrease in the predictive capability for positive cases.

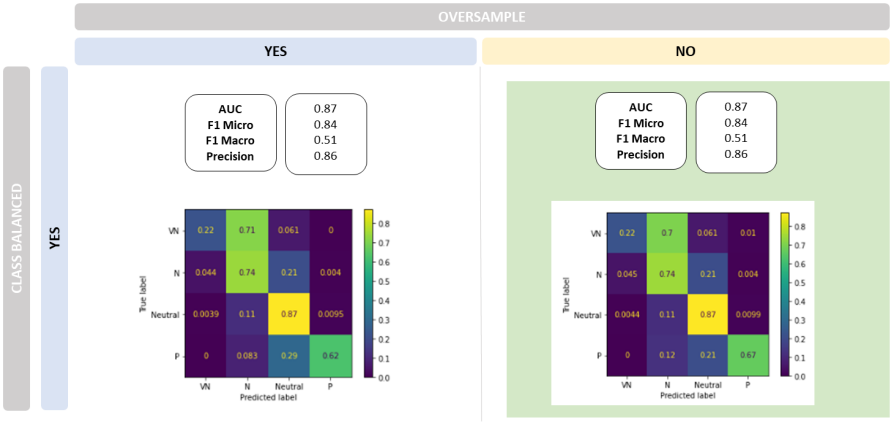


Figure 4.14: Impact of class balanced parameter

Concluding the tests on the multinomial logistic model, the next step was to assess the impact of including the features number of words, number of letters, number of vowels, and lexical diversity in the model, and determine which of these

four features or combination of features yielded the best results. After conducting the tests, it was observed that the best result was achieved when only the lexical diversity feature was included. Thus, concluding the analysis on this model, a total of 7286 features were selected. Additionally, an adjustment was made to the hyperparameter *max_iter* in order to optimize its value. Finally, with *max_iter* = 50, the best result obtained for this model can be found in Figure 4.15.

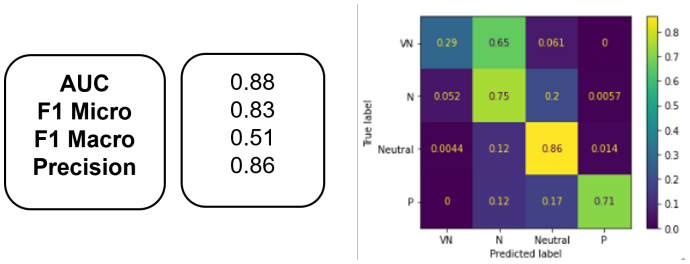


Figure 4.15: Best results - ML

After evaluating the initial three models, it was decided to further test the XGBoost model. For this purpose, the features that yielded the best results in the multinomial logistic model were considered, specifically, 7286 features. For this model, a similar test was conducted as in the case of the multinomial logistic model, regarding the use of the oversampling technique and the class balanced hyperparameter.

The results of the test, presented in Figure 4.16, showed that, similar to the ML model, the results are better when no oversampling is performed but the class balanced parameter is activated. However, the best results obtained with XGBoost are not as good as those obtained with multinomial logistic, and therefore, the latter model will continue to be considered as the better one.

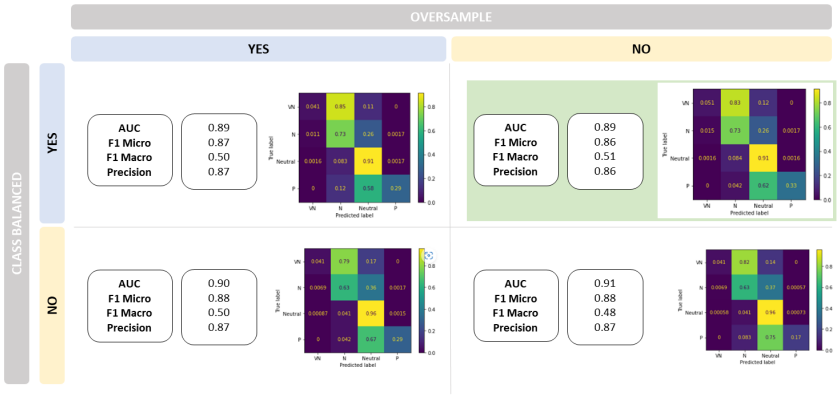


Figure 4.16: XGBoost

To conclude the analysis of proposed models, the ordinal regression model was also tested. For this model, the same set of 7286 variables was considered, and after

conducting several tests, the best result was obtained when applying the oversampling technique to the minority class. Once again, similar to the XGBoost model, the OR model did not achieve results as good as the multinomial logistic model, as can be seen in Figure 4.17.

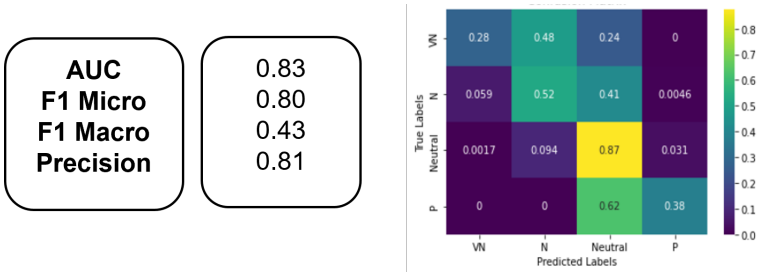


Figure 4.17: Ordinal Regression

Now, let's analyze the results obtained when using pre-trained models in English. For the three models used, as they were trained on English text, it became necessary to translate the project data that had already gone through the cleaning processes. For this purpose, it was utilized the *Translator* function from the *googletrans* library. It is important to note that due to the large amount of data, this function becomes quite complex and, consequently, computationally slow. Therefore, the test was only performed on 20,000 data points. Additionally, it should be mentioned that two of these models are designed to work with only 3 classes, namely negative, neutral, and positive. Therefore, it will be considered these 3 classes for the analysis by grouping very negative with negative and positive with very positive.

The first model tested was TextBlob. That is a powerful Python library for text processing and natural language processing tasks. When providing a message to this model, it will return a polarity associated with that message. The polarity can range from 0 to 1, where values close to 0 indicate negative polarity and values close to 1 indicate positive polarity.

Thus, an analysis was conducted to understand whether the data classified as negative exhibited lower polarity, if the data classified as neutral had an intermediate polarity, and finally, if the data classified as positive showed higher polarity.

It becomes evident, by Figure 4.21, that this model is not well suited for the given data, as it assigns intermediate polarities to the majority of negative data and low polarities to the majority of positive data.

The second pre-trained model used was Vader. Vader is a popular sentiment analysis tool provided by the $NLTK$ library. This model calculates the probabilities of a message being negative, neutral, or positive and then classifies it into the class

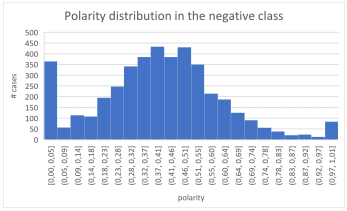


Figure 4.18: Negative class

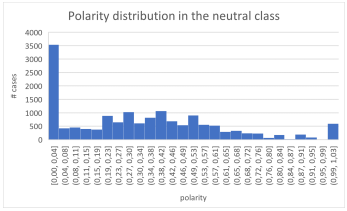


Figure 4.19: Neutral class

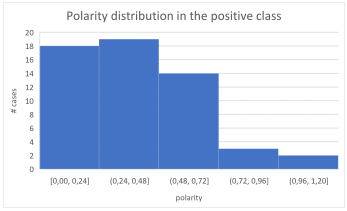


Figure 4.20: Positive class

Figure 4.21: Polarity distribution by class

with the highest probability.

The results of applying this model to the project data demonstrate, as can be seen in Figure 4.22, excellent performance in classifying neutral cases but very weak performance in classifying negative and positive messages. Such outcomes are unsatisfactory for the business as the main objective is not to identify neutral sentiments but rather more extreme sentiments such as negative and positive ones. This limitation suggests that the Vader model may not be the most suitable choice for accurately capturing and distinguishing strong emotions in the dataset.

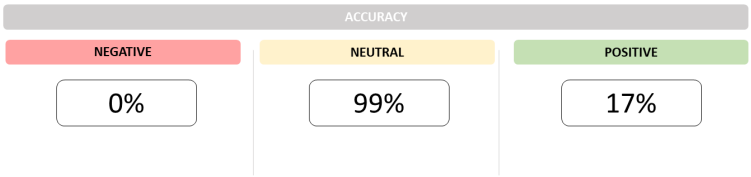


Figure 4.22: Vader

To conclude this section of results, the last pre-trained model to be evaluated is the Twitter Roberta Base Sentiment Latest model. This model, trained on a large corpus of twitter data, aims to classify sentiment in tweets. Similar to the previous models, it assigns probabilities to each message for being negative, neutral, or positive, and then assigns it to the class with the highest probability.

Contrary to the Vader model, the Twitter Roberta Base Sentiment Latest model exhibits a strong ability to classify negative and positive sentiments, but a lower capability in identifying neutral sentiment. This type of result, showed in Figure 4.23 is considered good for the business, as it can effectively capture negative sentiments. However, due to its lower accuracy in identifying neutral cases, there is a risk of misclassifying neutral cases as negative, which may not be ideal for the business as it would be considering cases as negative when they are not.

However, this model demonstrates a high accuracy in classifying negative and positive cases and enables the retailer to potentially prioritize neutral feedback with

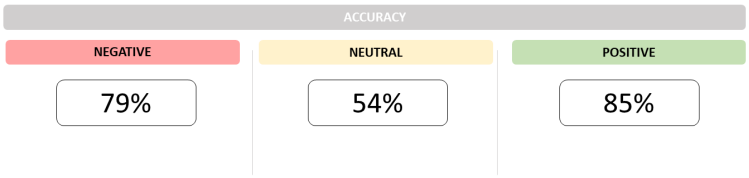


Figure 4.23: Twitter Roberta Base Sentiment Latest

a slight more tendency to be negative than regular neutral cases, given the high representation of neutral cases on the database. Therefore, similar to the classical multinomial logistic model, it is considered a potentially useful model for sentiment analysis by the customer support team of the Portuguese retailer.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The present study aimed at assessing various machine learning models, both proposed and pre-trained, in the domain of sentiment analysis. The ultimate goal was to find the most efficient model for the customer support team of a Portuguese retailer, such that it can more effectively handle and understand customer sentiment.

The analysis began with testing initial models such as naive Bayes, random forests, and multinomial logistic, with an early finding indicating that the multinomial logistic regression model outperformed the others in terms of AUC and precision values. This model exhibited a particularly strong predictive capacity in handling positive and neutral classes, and a satisfactory performance in predicting negative data.

Various data treatment methods and feature engineering techniques were employed in the course of the analysis to further optimize the model performance. The addition of certain features, namely, SS, PS, and NS, as well as the removal of less frequent words from the feature set, showed improvements in model performance. Including the features NA and LD also led to an enhancement in the AUC score.

In terms of handling imbalanced data, activating the 'class balanced' parameter proved beneficial in enhancing the model's performance, notably in the negative and very negative classes. However, applying both oversampling and the class balanced parameter concurrently resulted in a slight decline in the predictive capacity for positive cases.

The final set of features for the multinomial logistic model totalled 7286, after a thorough feature selection process. Hyperparameter tuning was also performed, resulting in an optimal value of 50 for the *max_iter* parameter.

The XGBoost and ordinal regression models were also tested, but neither yielded results as good as the multinomial logistic model. For these models, the same optimized set of features as used in the multinomial logistic model was considered.

The final part of the study involved assessing the performance of various pre-trained models. While these models were originally trained on english text, they were adapted to handle the portuguese language data in the project. Out of the three models evaluated - TextBlob, Vader, and Twitter Roberta Base Sentiment Latest - the latter demonstrated the highest accuracy in classifying negative and positive cases, thus making it a potentially useful tool for the customer support team.

However, it is important to note that the Twitter Roberta model showed a lower capacity to identify neutral sentiment, which could lead to misclassifications, particularly classifying neutral cases as negative. This would need to be taken into consideration when using this model in practice.

In conclusion, our analysis suggests that the multinomial logistic model and the Twitter Roberta Base Sentiment Latest model are the most suitable for the task of sentiment analysis for the portuguese retailer's customer support team. These models demonstrated high predictive capabilities in classifying negative, neutral, and positive sentiment, which is essential for effectively capturing and responding to customer sentiment. The key findings from this analysis contribute to a better understanding of sentiment analysis in customer service applications.

5.2 Future Work

While our results offer important insights, they also present opportunities for further research.

Considering that the training data used was previously classified, an opportunity for improvement is identified here, since, for future work, a business expert could be used to reclassify messages that, during this study, were found to be misclassified, due, for example, to the customers' use of irony. Another aspect that can be considered for future work is the increase of data, in order to make up for the reduced number of very positive cases and, thus, not requiring the merger of the positive class with the very positive class.

Refining the models and even applying ensemble techniques between models, for example between the two best models considered in this project, are also experiments that may be considered for even greater predictive performance.

Finally, Senti-Lex-PT can be changed, making it an even more diverse dictionary, adding cultural nuances and even words related to the retail sector.

Bibliography

[Wankhade et al. 2020] Wankhade, M., Rao, A.C.S. and Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artif Intell Rev*, (55), 5731–5780.

[Mowlaei et al. 2020] Mowlaei, M. E., Abadeh, M. S., and Keshavarz, H. (2020). Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications*, (148), 113–234.

[Kumar and Uma 2021] Kumar, A., and Uma, G. (2021). Intelligent sentinet-based lexicon for context-aware sentiment analysis: optimized neural network for sentiment classification on social media. *The Journal of Supercomputing*, 1 – 25.

[Ruffer et al. 2020] Ruffer, N., Knitza J., Krusche M. (2020). Covid4Rheum: an analytical twitter study in the time of the COVID-19 pandemic. *Rheumatol Int* , **40**(12), 2031–2037.

[Park et al. 2020] Park, H.W., Park, S., Chong, M. (2020). Conversations and medical news frames on twitter: infodemiological study on covid-19 in South Korea. *J Med Internet Res*, **22**(5), e18897.

[Cortis and Davis 2021] Cortis, K., Davis, R. (2021). Over a decade of social opinion mining: a systematic review. *Artif Intell Rev*, (54), 4873–4965.

[Arora et al. 2021] Arora, A., Chakraborty P., Bhatia M., Mittal P. (2021). Role of emotion in excessive use of twitter during COVID-19 imposed lockdown in India. *J Technol Behav Sci*, **6**(2), 370 – 377.

[Ahmad et al. 2019] Ahmad, S., Asghar M.Z., Alotaibi F.M., Awan I. (2019). Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Hum Centric Comput Inf Sci*, **9**(1), 1 – 23.

[Subhashini et al. 2021] Subhashini, L., Li, Y., Zhang, J., Atukorale, A.S., Wu, Y. (2021). Mining and classifying customer reviews: a survey. *Artif Intell Rev* , (54), 6343–6389.

[Pereira 2021] Pereira, D.A. (2021). A survey of sentiment analysis in the Portuguese language. *Artif Intell Rev* 54, 1087–1115.

[Ranchhod et al. 1999] Ranchhod, E., Mota, C., Baptista, J. (1999). A Computational Lexicon of Portuguese for Automatic Text Parsing. *SIGLEX99*: Standardizing Lexical Resources, w/pp. Association for Computational Linguistics.

[Barreiro et al. 2015] Barreiro, S., Sousa-Silva and Tagnin (eds.) Linguística, Informática e Tradução: Mundos que se Cruzam, *Oslo Studies in Language*, **7**(1), 425– 438

[Thet et al. 2010] Thet, T. T., Na, J.C., Khoo, C.S. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *J Inf Sci* , **36**(6), 823–848.

[Al Amrani et al. 2018] Al Amrani, Y., et al. (2018). Title of the Article. *Journal Name*, **Volume**(Issue), PageRange.

[Hassonah et al. 2020] Hassonah M.A., Al-Sayyed R., Rodan A., Al-Zoubi A.M., Aljarah I., Faris H. (2020b). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowl. Base Syst.*, (192), 105 – 353.

[Chang et al. 2020] Chang, J.R., Liang, H.Y., Chen, L.S., Chan,g C.W. (2020). Novel feature selection approaches for improving the performance of sentiment classification. *J Ambient Intell Humaniz Comput* , 1 – 14.

[Hosmer et al. 2013] Hosmer, D., Lemeshow, S., Sturdivant, R. (2013). Applied Logistic Regression, 3rd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA.

[Osborne 2017] Osborne, J. (2017). Best Practices in Logistic Regression; SAGE Publications, Ltd.: Thousand Oaks, CA, USA.

[Pituch et al. 2015] Pituch, K., Stevens, J. (2015). Applied Multivariate Statistics for the Social Sciences: Analyses with SAS and IBM's SPSS, 6th ed.; Taylor and Francis: Oxfordshire, UK.

[Hensher and Stopher 2021] Hensher, D., Stopher, P. (2021) Behavioural Travel Modelling; Taylor and Francis Inc.: London, UK.

[Ashqar et al. 2021] Ashqar, H., Shaheen, Q., Ashur, S., Rakha, H. (2021). Impact of risk factors on work zone crashes using logistic models and Random Forest. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–21 September 2021.

[Long 1997] Long, J. (1997). Regression Models for Categorical and Limited Dependent Variables, 1st ed.; SAGE Publications: Thousand Oaks, CA, USA.

[Rish 2001] Rish, I. (2001). An empirical study of the naive Bayes classifier. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 3(22), 41-46.

[Hand and Yu 2001] Hand, D. J., Yu, K. (2001). Idiot's Bayes: not so stupid after all? *International Statistical Review*, 69(3), 385-398.

[Mitchell 1997] Mitchell, T. M. (1997). Machine Learning. McGraw Hill. Chapter 3: Decision Tree Learning.

[Lewis 1998] Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In Proceedings of the 10th European Conference on Machine Learning (ECML 1998) (pp. 4-15).

[Agresti 2010] Agresti, A. (2010). Analysis of Ordinal Categorical Data (2nd ed.). Wiley.

[McCullagh 1980] McCullagh, P. (1980). Regression models for ordinal data, *Journal of the Royal Statistical Society*: Series B (Methodological), 42(2), 109–142.

[Vapnik 1995] Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc.

[Cortes and Vapnik 1995] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[Breiman 2001] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[Breiman and Cutler 2004] Breiman, L., Cutler, A. (2004). Random Forests - Machine learning. Retrieved from `https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`.

[Chen and Guestrin 2016] Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM.

[Hastie et al. 2009] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York, Springer.

[Almeida et al. 2023] Almeida, C., Castro, C., Braga, A. C., and Freitas, A. (2023). Hybrid Sentiment Analysis of Portuguese Retail Customer Feedback: A Comprehensive Approach Leveraging Lexicon and Machine Learning Techniques. MDPI. https://www.mdpi.com/journal/applsci

Appendix A

Wordclouds

Wordclouds can be generated based on the desired amount of information to be extracted. Smaller wordclouds can provide a more superficial overview of the content, while larger wordclouds can offer more detailed information.

In this appendix, two examples of wordclouds with ten and twenty-five words are presented for each group of messages, divided by sentiment class.

Positive

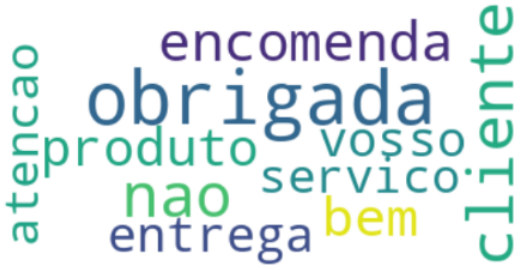


Figure A.1: Ten words

Figure A.2: Twenty-five words

Figure A.3: Wordcloud of the positive class

Neutral

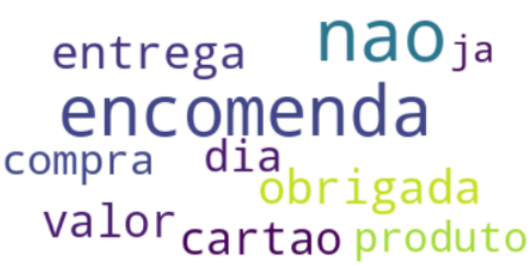


Figure A.4: Ten words

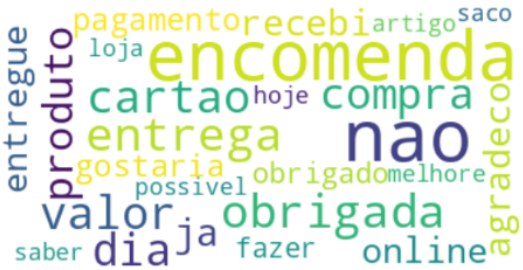


Figure A.5: Twenty-five words

Figure A.6: Wordcloud of the neutral class

Negative

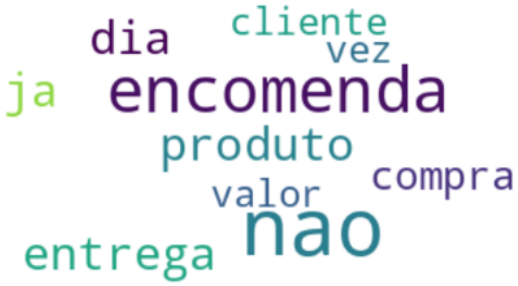


Figure A.7: Ten words

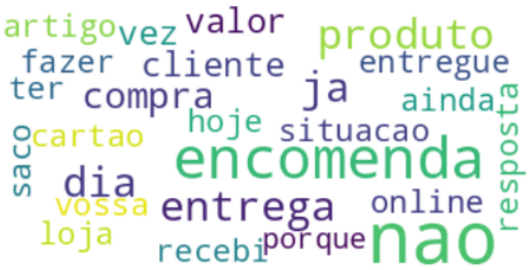


Figure A.8: Twenty-five words

Figure A.9: Wordcloud of the negative class

Very Negative

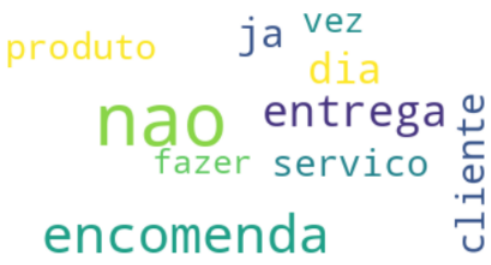


Figure A.10: Ten words

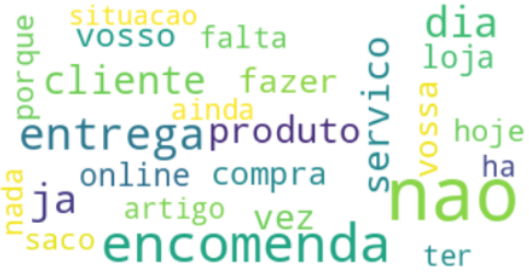


Figure A.11: Twenty-five words

Figure A.12: Wordcloud of the very negative class

Appendix B

N-Grams

Just like wordclouds, n-grams can also be generated based on the desired amount of information to be extracted.

Here, an example is presented that shows the five most frequent bi-grams and tri-grams for each sentiment class.

Positive

```
Out[33]: [(('desde', 'ja'), 6),
 (('bom', 'trabalho!'), 6),
 (('ola', 'ambito'), 5),
 (('ambito', 'projeto'), 5),
 (('projeto', 'folhetos'), 5)]
```

Figure B.1: Bi-grams

```
Out[34]: [(('ola', 'ambito', 'projeto'), 5),
 (('ambito', 'projeto', 'folhetos'), 5),
 (('projeto', 'folhetos', 'personalizados'), 5),
 (('folhetos', 'personalizados', 'recebemos'), 5),
 (('personalizados', 'recebemos', 'seguinte'), 5)]
```

Figure B.2: Tri-grams

Figure B.3: N-grams of the positive class

Neutral

```
Out[31]: [(('gostaria', 'saber'), 2006),
 (('nao', 'recebi'), 1408),
 (('nao', 'consigo'), 1256),
 (('ainda', 'nao'), 1207),
 (('encomenda', 'nao'), 1093)]
```

Figure B.4: Bi-grams

```
Out[30]: [(('obter', 'outlook', 'android'), 540),
 (('ainda', 'nao', 'recebi'), 342),
 (('venho', 'meio', 'solicitar'), 312),
 (('encomenda', 'manuais', 'escolares'), 245),
 (('enviado', 'iphone', 'dia'), 243)]
```

Figure B.5: Tri-grams

Figure B.6: N-grams of the neutral class

Negative

```
Out[36]: [(('ainda', 'nao'), 727),
 (('venho', 'meio'), 579),
 (('nao', 'recebi'), 561),
 (('fiz', 'encomenda'), 467),
 (('encomenda', 'nao'), 440)]
```

Figure B.7: Bi-grams

```
Out[38]: [(('nao', 'primeira', 'vez'), 182),
 (('ainda', 'nao', 'recebi'), 177),
 (('obter', 'outlook', 'android'), 144),
 (('ja', 'nao', 'primeira'), 127),
 (('venho', 'meio', 'reclamar'), 102)]
```

Figure B.8: Tri-grams

Figure B.9: N-grams of the negative class

Very Negative

```
Out[41]: [(('venho', 'meio'), 44),
 (('un', 'un'), 37),
 (('ja', 'nao'), 35),
 (('vosso', 'servico'), 31),
 (('fiz', 'encomenda'), 28)]
```

Figure B.10: Bi-grams

```
Out[40]: [(('alvaro', 'pinto', 'enviada'), 9),
 (('pinto', 'enviada', 'junho'), 9),
 (('enviada', 'junho', 'ajudacontinente'), 9),
 (('junho', 'ajudacontinente', 'cc'), 9),
 (('ajudacontinente', 'cc', 'alvaro'), 9)]
```

Figure B.11: Tri-grams

Figure B.12: N-grams of the very negative class