UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Análisis Automático de Experiencia de Cliente en los Canales de Atención Digitales

Máster Universitario en Ingeniería Informática

Autor: ZAYAT MATA, Ana

Tutora: CARRO SALAS, Rosa María

Departamento de Ingeniería Informática

FECHA: Septiembre, 2019

Análisis Automático de Experiencia de Cliente en los Canales de Atención Digitales

AUTOR: Zayat Mata, Ana

TUTOR: Carro Salas, Rosa María

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Septiembre, 2019

Resumen

Muchas empresas que trabajan con usuarios finales quieren conocer las opiniones de sus clientes sobre sus productos para poder mejorar la experiencia de usuario y mantenerlos dentro de la empresa. Para conseguir las opiniones de los clientes, estas empresas les solicitan que rellenen una serie de formularios en los que indican, entre otros, su grado de satisfacción con el producto o con la empresa. Esos formularios también suelen tener un apartado de texto libre en el que el cliente puede expresar sus opiniones, quejas o sugerencias.

Una vez tienen los datos de los formularios, las empresas necesitan procesarlos para poder obtener información relevante para ellos. El apartado de texto libre es muy importante para la empresa, ya que puede contener información muy útil para entender los problemas que pueda tener un producto o servicio, o incluso la empresa. El problema es que, en muchas empresas, procesan este campo manualmente (hay empleados encargados de leer cada comentario y anotar a qué se refiere), con lo que invierten mucho tiempo y recursos en leer y categorizar todos los comentarios recibidos de los clientes.

Este es el caso de Vodafone, empresa en la que cada mes se reciben alrededor de 2400 comentarios que los clientes envían a través de formularios online y que se han de categorizar para conocer el tipo de servicio, producto o aspecto de la compañía que motiva el comentario. Con el objetivo de agilizar esta tarea de categorización de comentarios, Vodafone incluyó, como uno de los retos a abordar en el Programa Vodafone Campus Lab, el proyecto “Análisis Automático de Experiencia de Cliente en los Canales de Atención Digitales”. La autora de esta memoria fue la beneficiaria de la beca asociada a dicho reto, beca con la que Vodafone ha financiado la realización de este trabajo fin de máster, centrado por tanto, en resolver el problema descrito y contribuir así a la categorización de comentarios de forma automática.

En este trabajo se ha explorado la utilización y combinación de los mejores métodos disponibles actualmente para el procesamiento y la clasificación automática de textos, con el objetivo de categorizar los comentarios de los clientes de Vodafone. Se ha trabajado con diversos algoritmos y con más de 16.680 comentarios reales. Se ha desarrollado una solución informática que, incorporando el método que ha ofrecido mejores resultados para los datos disponibles, da soporte a la categorización automática de comentarios. Este software está siendo evaluado actualmente en Vodafone y está prevista su próxima incorporación al conjunto de herramientas que utiliza la empresa en este ámbito.

Palabras clave

Minería de opinión, categorización de textos, procesamiento de lenguaje natural, clasificación automática, satisfacción del cliente, entorno real, Vodafone.

Abstract

Many companies who work with final users want to learn their users' opinions on their products to be able to improve the user experience and keep them in the company. In order to obtain these opinions, these companies ask their users to fill in a series of forms where they can input, among other things, their satisfaction level with the product and the company. These forms usually also have a free-text field where the client can express their opinions, complaints and suggestions.

Once these forms are filled in, the companies need to process them to be able to obtain relevant information for them. The free-text field is very important for the company because it might contain a lot of very useful information to understand the problems a product or service, or even the company might have. The problem is that, in many companies, they process this field manually (there are employees in charge of reading each and every comment and write down what they are referring to), which means they invest a lot of time and resources to read and categorize all these comments received from the clients.

This is the case for Vodafone, company who, in a month, receive around 2400 comments that the clients send via the online forms and that need to be categorized to learn the type of the company's service, product or other specific aspect is motivating the comment. With the objective of streamline this task of comment categorization, Vodafine included, as one of the challenges to overcome in the Program Vodafone Campus Lab, the project "Análisis Automático de Experiencia de Cliente en los Canales de Atención Digitales". The author of this thesis was the recipient of the scholarship associated to said challenge, scholarship with which Vodafone has financed the development of this master's thesis, focused therefore, in solving the problem described and this way contribute to the categorization of clients' comments automatically.

This project has explored the use and combination of some of the best methods for text processing and automatic categorization that exist nowadays, with the objective of categorizing the clients' comments Vodafone has. Work has been various algorithms and with more than 16,680 real comments. This software is currently being evaluated in Vodafone and it is planned to be incorporated into the collection of tools the company used in this area.

Keywords

Opinion mining, text categorization, natural language processing, automatic classification, client satisfaction, real environment, Vodafone.

Acknowledgements

First and foremost, I want to thank my tutor Rosa for believing in me and helping me through this whole process.

I also want to thank Carlos and Raquel for giving me the opportunity to work with a company such as Vodafone and helping me with everything I asked for, even if it was last minute. Thank you for everything, this project would not be possible without you.

Finally, I want to thank my family and friends, who supported and encouraged me throughout my years at the EPS and through the process of developing and writing this thesis.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

1.1 MOTIVATION

Currently, most companies and entities who work with final users need to know their clients' satisfaction level towards their products as well as their opinions. To be able to get this information, they ask their clients to fill out a series of forms in which they can give their opinion in different ways. Even though many design their own questionnaires to be able to process them as automatically as possible, most add an open-ended field where users can express themselves and their opinion on a specific product, service or any aspect related to the company or entity using free text.

By using this method to get clients' opinions, companies can obtain very specific information on what their clients are referring to when they fill in their satisfaction level and the reasons behind this opinion. This is very useful, but they face a common problem: the difficulty of categorizing these opinions to be able to use them to better understand their clients' problems.

Many companies spend a lot of money and human resources to analyse these texts manually, which basically means having to read every comment written by a client, identify the concrete problem they are referring to (categorize them) and associate the satisfaction level, expressed by the client numerically, to the specific category it refers to, as detected in the comment. This is the case of companies such as Vodafone. Nevertheless, the advances made in the areas of natural language processing and machine learning have created techniques and algorithms that, without a doubt, can be exploited to try to solve the very challenging problem of obtaining the clients' opinions trough free text in a more automatic way.

The need to find more automatic solutions to this problem is real, as shown, for example, in the goals of the scholarships some companies offer to enrich and better their products and customer service with research results and new solutions created. This was the case of the program Vodafone Campus Lab [1], in which Vodafone offered 20 scholarships to masters and doctorate students to develop 20 projects, one of them focused on automating the analysis of their clients' satisfaction. This scholarship was awarded to the author of this thesis and is the reason for this work, aimed at exploring the possibility of using natural language processing and machine learning to solve the company's problem of having to categorize their clients' comments manually.

1.2 OBJECTIVE

The objective of this work is to offer an automatic solution to analyse and categorize users' comments used to express their opinions about a company and its products. This objective can be divided into the following smaller objectives:

- Investigate the current state of the art in terms of use of natural language processing techniques in this context.
- Analyse the possibilities machine learning offers to work with and categorize texts.
- Analyse the data provided by the company to have a better understanding of the information available and its structure.
- Explore the usage and combination of techniques and algorithms to process the data, with the goal of designing an effective solution for the problem.
- Design and implement a unique solution for the problem.
- Present and deploy the solution in the company. The solution will be integrated with the software in use at the moment to create a complete solution.
- Analyse the advantages and limitations of the current solution and propose future lines of work.

1.3 DOCUMENT'S ORGANIZATION

This document has the following chapters:

- **Chapter 1, Introduction:** this chapter explains the motivation and the objective of this master's thesis

- **Chapter 2, State of the Art:** this chapter gives a current overview of how natural language is processed, what machine learning algorithms are currently being used and the most used opinion mining techniques nowadays.

- **Chapter 3, Analysis:** this chapter presents the analysis of the data available for this work , as well as the decisions taken regarding the text processing methods and text categorization algorithms selected for this work.

- **Chapter 4, Testing and Results:** this chapter shows the results of all combinations of text processing and categorization methods tested for this project. The final results are also described.

- **Chapter 5, Technology Used and Integration:** this chapter describes the technologies used to develop this project and the integration of the project in the company.

- **Chapter 6, Conclusions and Future Work:** this chapter presents the conclusions, the acquired knowledge and the possible future lines of action.

2 STATE OF THE ART

2.1 NATURAL LANGUAGE PROCESSING

Language, as defined in the Cambridge Dictionary [2], "is a system of communication consisting of sounds, words and grammar, or the system of communication used by people in a particular country or type of work". Another definition is, "in computer programming, a language is a system of writing instructions for computers". These two definitions show the difference between the computer languages and what is known as "natural language". This natural language is what we use to communicate between us, talking or writing but it is not easy for a computer to understand this language. This is the reason why, for many years now, we have tried to create methods to process this natural language into something the computer can understand. These technologies we created to help computers understand our natural language is Natural Language Processing or NLP.

Language has been described as a system since the early 1900s by Professor Ferdinand de Saussure, when he described meaning as being created by the relationships within a language, and that communication is achieved by sharing a language system. In 1950, Alan Touring designed a test to determine if a machine could "think". For a computer to pass this test, a human has to be able to communicate with it and think they are both humans. These events helped get the inspiration for the idea of Artificial Intelligence and Natural Language Processing. NLP has had its ups and downs through the years but in the recent years more and more companies are making use of it for their products [3].

NLP has the objective of reading, deciphering and understanding the natural language in a way that can be used later on [4]. With such a broad goal, it can be used for many common applications like Google Translate, Microsoft Word, Grammarly, OK Google, Alexa, Siri, Cortana, and many more. These are different ways of using NLP since some of them work with written text, some with speech, and others a mix of both. NLP can be used for many different things like text-to-speech conversion or vice versa, sentiment analysis, contextual extraction, translations, … [3], but in this thesis I will focus on content categorization.

The main NLP methods used nowadays are Bag of Words, TF-IDF and Word2Vec [5].

- Bag of Words [6, 7, 8]:

Counts the number of times a word appears in a text. It doesn't care about the word order or relevance. It allows the comparison between texts and it helps process the text for machine learning.

Figure 2-1 shows the representation of BoW on an example text:

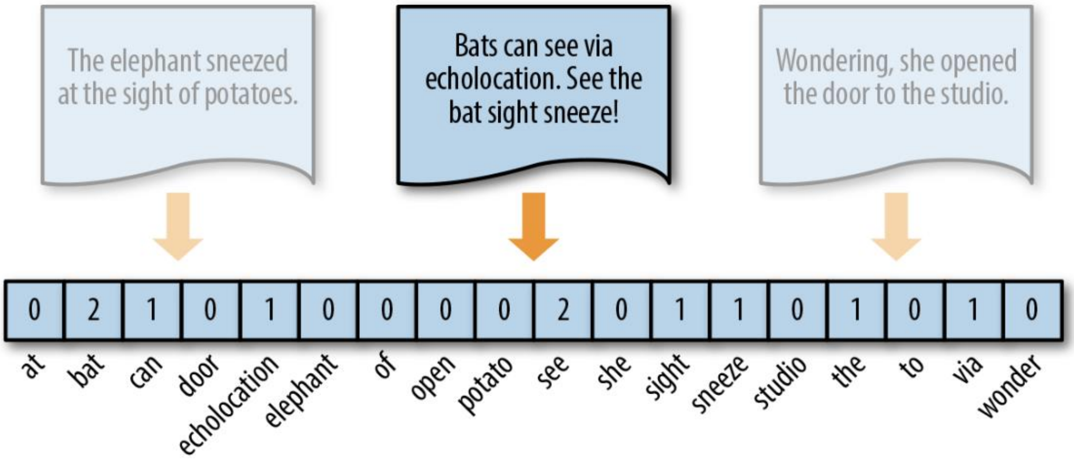

Figure 2-1: Bag of Words Example [45]

- TF-IDF [8, 9, 10]:

Term-Frequency-Inverse Document Frequency. It gives each word a weight depending n their relevance. First it measures the times the word appears in the text, but the more texts a word appears in the less relevant and so the lower the score.

Figure 2-2 shows the formula used for the TF-IDF calculation:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Figure 2-2: TF-IDF Formula [8]

- Word2Vec [11, 12]:

Created by Google, it creates a vector for each word, taking into account their meaning, their relevance and their frequency.

Figure 2-3 shows two diagrams of the two methods used by Word2Vec to train the model:

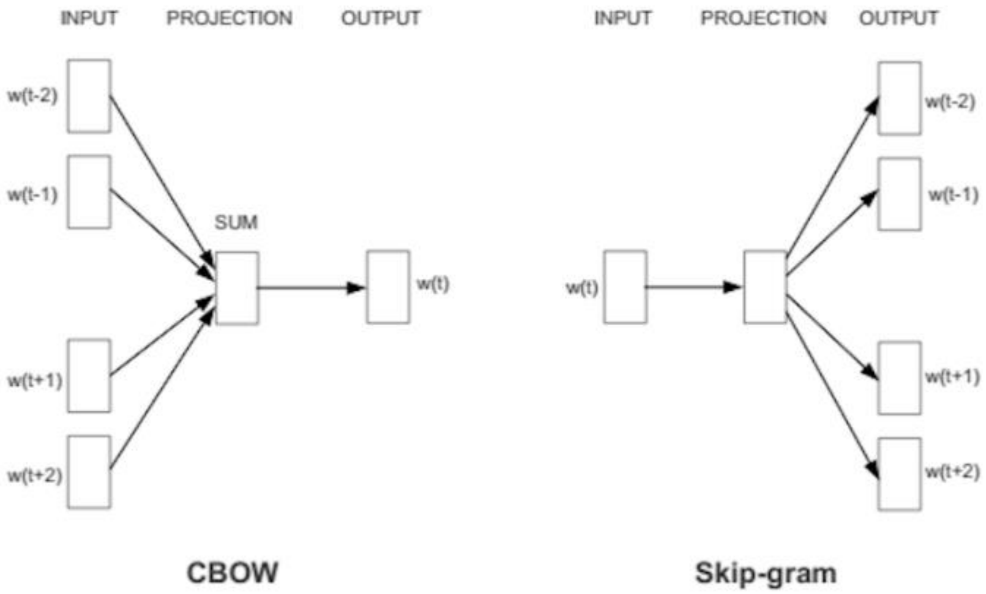

Figure 2-3: Word2Vec Diagram [8]

In section 2.3 there will be more examples of these techniques being used for opinion mining.

2.2 MACHINE LEARNING

"Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data and information. In machine learning computers don't have to be explicitly programmed but can change and improve their algorithms by themselves." [13]. As seen before, Alan Turing created a test to decide whether a computer could imitate a human well enough to pass as a human. This was the beginning of machine learning. Since that moment, many have created computers and programs that can learn and evolve, whether it was playing checkers, talking or navigating obstacles. In 2011 a machine beat its human opponents at Jeopardy, in 2014 a face recognizing software was launched, and even cars can now be autonomous and don't need a driver [13].

There are two main ways to classify data, using supervised or unsupervised algorithms. For the first ones, the training data needed has to already be classified for them to learn from it. The second ones do not need this since they find similarities in the data to classify it. Since the data is already classified, I will focus on supervised machine learning. Since the data has also text features, here are some of the algorithms used for text classification:

- Decision Tree Classifier [14, 15]

This classifier uses decision trees to divide the data and later put it in categories. This algorithm uses entropy (degree of randomness) and information gain (calculates which division will give better results), together with recursion, to be able to classify the data by dividing it until there is nothing left to divide, or a certain criterion is met.

- Naïve Bayes [16]

This classifier applies the Bayes Theorem to classify data using the probabilities of each feature to appear for each category and choosing the one with the highest result. It is called naïve because all features are considered independent from each other.

- Stochastic Gradient Descent Classifier [17]

This classifier uses gradient descent as an optimization technique. This means it uses a function to iteratively find values to minimize the cost function as much as possible. Stochastic is a process linked to a random probability, so this algorithm uses randomly selected samples of the data for each iteration which makes this algorithm faster since it does not need to use all the dataset to reach a conclusion.

- Random Forest [18, 19]

Similar to decision tree classifier, this algorithm uses a number of decision trees and groups their outputs to obtain results. This is a very robust model, but it is a bit of a black box.

- Extreme Gradient Boosting [20]

Also using decision trees, this algorithm uses a series of fixed sized trees as its base. This algorithm was designed to improve trees' speed and performance by boosting them. Boosting means creating new models to correct the errors existing models made, they are added sequentially until there are no improvements left and then they are all added together to give the final prediction.

- Logistic Regression [21]

This algorithm makes use of probabilities to predict the classification. It is very similar to a linear regression, but it uses a more complex function to calculate the cost. This function is called Sigmoid function or logistic function. In logistic regression the cost function is limited between 0 and 1, unlike the linear one, and it maps every input into a value between 0 and 1 to obtain probabilities as the predictions.

- Neural Network

Trying to copy the way animal brains work this type of classification uses a number of nodes or neurons, in sequential layers, that process the input they have and send it to the next layer or group of neurons to process again. Depending on the different layers the network can perform different transformations to the input to obtain a different output. There are many different neural networks but, after some investigation [22, 23, 24], these are the main ones:

    - CNN (Convolutional Neural Network) [25]: This method is very common for analysing images but has proven to be quite useful when classifying text. It uses various layers whose neurons are all connected to the one in the next layer.
    - RNN (Recurrent Neural Network) [26]: Here, the connections between neurons create a directed graph. It is able to use an internal state or memory to process inputs in time sequence.

Machine learning is very powerful and has many uses but for this thesis I will focus on analysing texts and categorizing them. This is generally known as Opinion Mining.

2.3 OPINION MINING

Data Mining is the name given to the process of finding relevant information in large data sets with the help of machine learning and other useful data processing methods.
Opinion Mining is a branch of data mining in which the relevant information wanted involves users' opinions and sentiments. Normally, for opinion mining, the useful information is in texts so there is a need to process these texts before introducing the data in the machine learning algorithm to learn.

Some lines of work in opinion mining use NLP techniques such as tokenization, word segmentation, part of speech, tagging and parsing [27]. Sometimes NLP is used together with dictionaries to identify sentiments [28]. Some others use non supervised classification methods to classify text depending on the sentiment found in them [29]. Some try to use a combination of different approaches such as combining a lexicon-based sentiment classifier with a machine learning based classifier [30]. And others summarize the users' opinions to get specific information and obtain sentiment polarity using part of speech tagging [31]. In [32], the authors analyse the mapping of opinion mining and sentiment analysis publications through the years. They conclude that there is significant growth in investigation in this area and it has spread worldwide unlike before when only a few countries published about this topic. They also conclude that there is more research on machine learning approaches (67.20%) tan on lexicon-based ones (27.15%) and document-level sentiment analysis is on the rise.

Some of the opinion mining works in the business domain, focused on knowing the clients opinions on products offered by a company are the following: in [42] they use Alchemy API to analyse tweets to obtain users' polarity towards a specific product; in [43] they use NLP and Naïve Bayes to determine user's reviews polarity about a product to be able to recommend new products; and in [44] they compiled a series of research papers on opinion mining being used to obtain information from mobile app stores' users and they mention some papers where they obtain 91% precisions using Statistical Analysis, 59% precision using Linear Discriminant Analysis, 59% accuracy using Naïve Bayes and Decision Trees and 83% accuracy using ranking.

3 ANALYSIS

3.1 DATA ANALYSIS

To begin this project, the company provided an excel file with all the data to be used. This document consisted of all the data the company obtains every time a client fills out and submits a questionnaire. Clients are normally asked to fill out questionnaires after they have finished a task or after solving a problem, and they appear in the web application as well as the mobile app.

The data available includes information on the page where the user was asked to fill in the questions, the user's satisfaction with the product, the degree to which the user would recommend the company, free text fields with the user's comments on the product and the company and whether they could fix the issue they had, and many more.

After a couple meetings with the people currently working on this data in the company, some columns were chosen as relevant to the categorization of the questionnaire. These columns are:

- Form Name: Shows the journey the user took to get to the form. The values can be either Web or Mobile and between Bill, Account or Usage.

- Recommendation: Shows the users' answer to the question "Would you recommend the company?". The values are numeric from 0 to 10.

- Resolution: Shows the answer to the question "Did you solve your problem?". Yes and No values.

- Satisfaction: Shows the users' satisfaction level with the company/product.

- Try other way: Gathers the text answers to the question "If your problem has not been solved, how would you try again?"

- NPS: Collects the information on whether the user is a promoter, neutral or a detractor of the company/product.

- Hard: Shows whether the users' recommendation score is too low or not. Values are Hard and Not Hard.

- Verbatim: Text field gathering the users' opinions and comments.

- Category: Includes all the categories the company created for the data.

The company provided various datasets with comments categorized. These datasets have a total of 16,680 rows, each with all the data obtained from the clients' questionnaires. Since the company is interested in keeping their clients happy and the users that are detractors (NPS) are the ones at risk of leaving the company, it was decided that this thesis would focus on categorizing these users' verbatim. When filtering the datasets to only work with this data the number of rows left is 8,893, about half of the data provided. After taking this into consideration, the information shown in figures 3-4, 3-5, and 3-6 was obtained.

Figure 3-4 shows the categories established by the company and the number of comments in each category. For example, there are 39 comments categorized in "mal funcionamiento: lentitud". As can be seen, there are two categories that have the most amount of comments ("quejas generals sobre Vodafone" and "sin comentario del cliente") and there are some that have very few ("mal funcionamiento: registro y logado"). This will influence the training of the algorithm since there might not be enough comments to learn from.

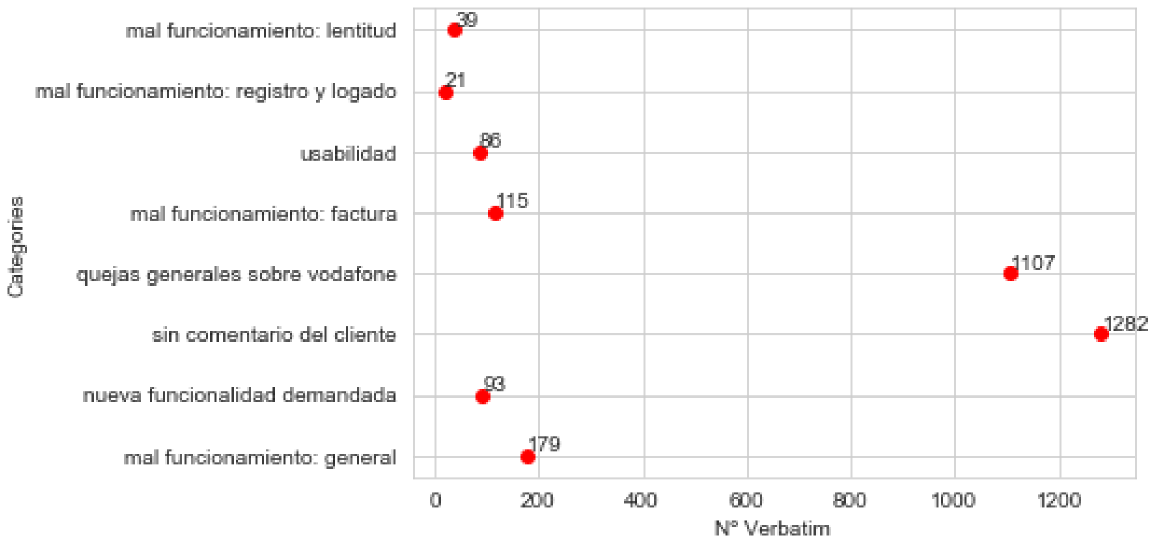

Figure 3-4: Number of Verbatim for Each Category

Figure 3-5 shows columns Recommendation, Resolution, Satisfaction and Hard and for each it shows the probability density of each different value these columns have. As can be seen, Resolution and Hard are binary since they only have 0 and 1 values, and Recommendation and Satisfaction have a larger density in the low values.

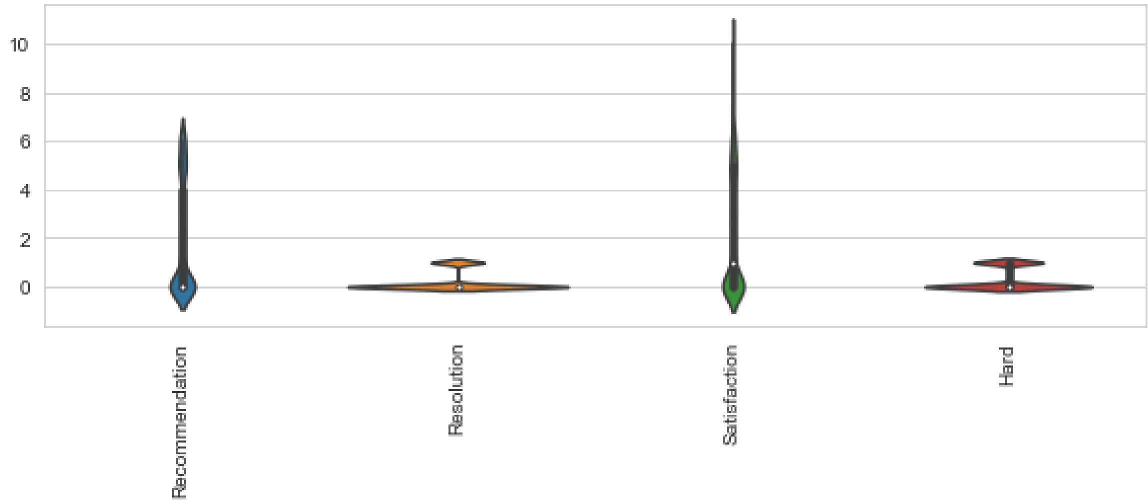

Figure 3-5: Numerical or binary columns' distribution

Figure 3-6 shows the different Form Names in the dataset and the number of comments created in each one. As can be seen, there are many clients accessing all three pages but the majority fill in the questionnaires after accessing the "View Bill and Pay" page.

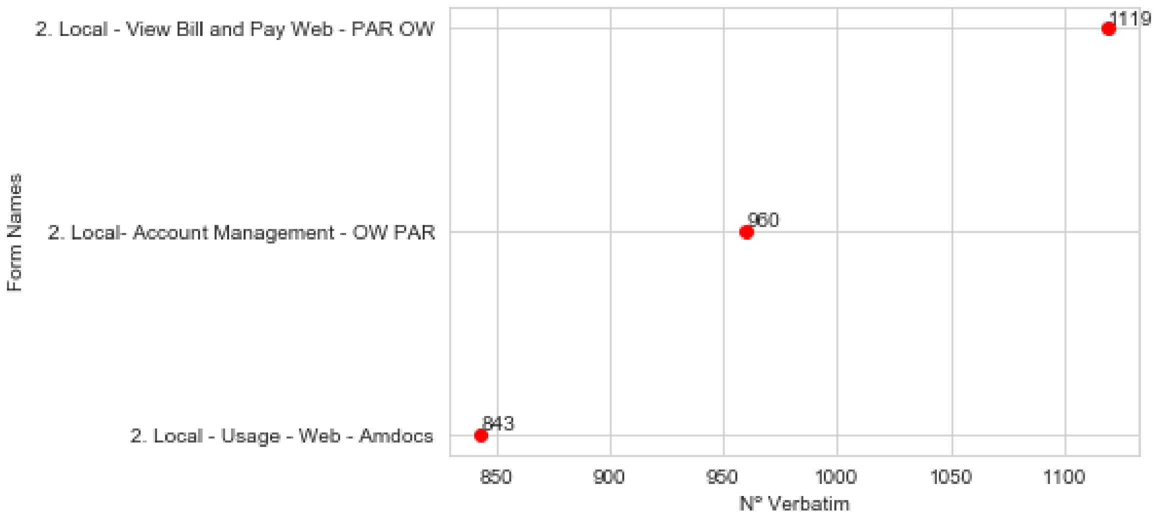

Figure 3-6: Number of Verbatim for Each Form

3.2 DATA PROCESSING AND ALGORITHMS

There are many ways to process data but, since the data used has text as some of its main features, it needs to be considered. As already seen in section 2.1, there are many ways to process text, but I have focused on the main three described in that section: Bag of Words, IF-IDF and Word2Vec. This was decided because all three have shown to obtain good results and I wanted to test them all and see how each can work with the data to try and solve the problem.

But before these can be implemented, the text has to be cleaned to be able to have real meaningful words. To clean the text, I have to first, eliminate all the non-alphabetic characters (here accentuated letters are included since the data is in Spanish). Second, all words must be in lowercase and, finally, I will have to eliminate the words that don't add any extra meaning like articles, conjunctions and prepositions, all these words are called stopwords.

Regarding automatic classification, there are many algorithms that can be used and since I could not try every single one, I decided to test the ones explained in section 2.2. because they are all very common and have proved to be useful for text classification (e.g, CNN and RNN in [22, 23, 24]; Naïve Bayes, SGD and Logistic Regression in [33]; Decision Tree in [34]; Random Forest in [35]; and XGB in [36]) and I want to try and see which one can obtain the best results for the data. The algorithms I used are: Naïve Bayes, Logistic Regression, Random Forest, Extreme Gradient Boosting, Stochastic Gradient Descent, Decision Tree, Convolutional Neural Network and Recurrent Neural Network.

To test the algorithms' results I divided the data into training and testing, using 5,971 rows to train the models and 2,922 to test them.

4 TESTING AND RESULTS

In this section all the different options tried will be described together with their results. For most algorithms I have used both Bag of Words and TF-IFD vectorizers, both counting frequencies of words, ngrams (group of n words, in our case 2-3 words), and characters (in this case 2-3 characters) so as to see which approach is the best for our dataset. For all classifiers the number of classes is the same, all 8 classes corresponding to the categories set by the company, so as to not deviate from the goal. For most of the algorithms' configurations I decided to use the default values from Scikit-learn, for the ones I did some changes I will explain in the corresponding subsection.

The figures included in the following subsections show information regarding the accuracy obtained by each algorithm, the precision with which each category was predicted, the percentage of recall sensitivity of each category (the correctly predicted divided by the total), the f1-score (mean between precision and recall) for each category and finally the number of comments for each category or support.

4.1.1 Naïve Bayes

For this algorithm I used all the columns mentioned in chapter 3 and processed the text columns using Bag if Words and TF-IDF, counting frequencies of words, n-grams and characters.

For the Bag of Words approach the results shown in figures 4-7, 4-8 and 4-9 were obtained:

```
accuracy 0.7532511978097194
                                       precision    recall  f1-score   support

                         usabilidad       0.68      0.33      0.44       115
          sin comentario del cliente       0.52      0.36      0.43       179
     quejas generales sobre vodafone       0.00      0.00      0.00        39
         mal funcionamiento: general       0.71      0.24      0.36        21
mal funcionamiento: registro y logado       0.80      0.09      0.16        93
         mal funcionamiento: factura       0.80      0.73      0.77      1107
       nueva funcionalidad demandada       0.74      0.99      0.85      1282
        mal funcionamiento: lentitud       0.83      0.06      0.11        86

                          micro avg       0.75      0.75      0.75      2922
                          macro avg       0.64      0.35      0.39      2922
                       weighted avg       0.74      0.75      0.72      2922
```

Figure 4-7: Naïve Bayes – Bag of Words – Words

accuracy 0.5852156057494866

	precision	recall	f1-score	support
usabilidad	0.63	0.23	0.33	115
sin comentario del cliente	0.60	0.14	0.23	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.40	0.19	0.26	21
mal funcionamiento: registro y logado	0.62	0.05	0.10	93
mal funcionamiento: factura	0.61	0.47	0.53	1107
nueva funcionalidad demandada	0.57	0.88	0.70	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.59	0.59	0.59	2922
macro avg	0.43	0.25	0.27	2922
weighted avg	0.57	0.59	0.54	2922

Figure 4-8: Naïve Bayes – Bag of Words – Ngrams

accuracy 0.6735112936344969

	precision	recall	f1-score	support
usabilidad	0.31	0.80	0.44	115
sin comentario del cliente	0.39	0.36	0.37	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.03	0.48	0.06	21
mal funcionamiento: registro y logado	0.21	0.45	0.29	93
mal funcionamiento: factura	0.91	0.46	0.61	1107
nueva funcionalidad demandada	0.94	0.95	0.94	1282
mal funcionamiento: lentitud	0.34	0.41	0.37	86
micro avg	0.67	0.67	0.67	2922
macro avg	0.39	0.49	0.39	2922
weighted avg	0.81	0.67	0.71	2922

Figure 4-9: Naïve Bayes – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-10, 4-11 and 4-12 were obtained:

accuracy 0.7152635181382615

	precision	recall	f1-score	support
usabilidad	0.00	0.00	0.00	115
sin comentario del cliente	0.00	0.00	0.00	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.00	0.00	0.00	21
mal funcionamiento: registro y logado	0.00	0.00	0.00	93
mal funcionamiento: factura	0.76	0.74	0.75	1107
nueva funcionalidad demandada	0.69	0.99	0.81	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.72	0.72	0.72	2922
macro avg	0.18	0.22	0.20	2922
weighted avg	0.59	0.72	0.64	2922

Figure 4-10: Naïve Bayes – TF IDF – Words

accuracy 0.5694729637234771

	precision	recall	f1-score	support
usabilidad	0.00	0.00	0.00	115
sin comentario del cliente	0.00	0.00	0.00	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.00	0.00	0.00	21
mal funcionamiento: registro y logado	0.00	0.00	0.00	93
mal funcionamiento: factura	0.71	0.38	0.50	1107
nueva funcionalidad demandada	0.53	0.97	0.69	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.57	0.57	0.57	2922
macro avg	0.16	0.17	0.15	2922
weighted avg	0.50	0.57	0.49	2922

Figure 4-11: Naïve Bayes – TF IDF - NGrams

accuracy 0.7984257357973991

	precision	recall	f1-score	support
usabilidad	0.78	0.40	0.53	115
sin comentario del cliente	0.47	0.09	0.16	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.00	0.00	0.00	21
mal funcionamiento: registro y logado	1.00	0.01	0.02	93
mal funcionamiento: factura	0.69	0.95	0.80	1107
nueva funcionalidad demandada	0.93	0.95	0.94	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.80	0.80	0.80	2922
macro avg	0.48	0.30	0.31	2922
weighted avg	0.76	0.80	0.75	2922

Figure 4-12: Naïve Bayes – TF IDF - Characters

Looking at these results we can see that for the Naïve Bayes algorithm, considering accuracy, the best approach is TF-IDF using 2-3 characters to count their frequencies with an accuracy of 0.798 against accuracies of 0.753 obtained in BoW with words or 0.715 obtained in TF-IDF with words. But if we consider the whole classification report we can see that there are come categories that are very well classified like "mal funcionamiento: registro y logado", with a precision of 0.93, but there are some that are never assigned, which means that the algorithm has not predicted a single comment for that category and is shown in the figure with a precision value of 0.00. This is a good result, almost 80% accuracy, but not the best overall because of these categories that are not used in the prediction.

4.1.2 Logistic Regression

For this algorithm I also used all columns from section 3 and processed the text columns using Bag if Words and TF-IDF. For the configuration of this algorithm I initialized it with 1 job (number of CPU cores for parallelization), $1e^5$ C (regularization strength), 'saga' solver (algorithm for optimization) and 5000 max iterations for the solver to converge. I tried with different versions and this was the one with the best results.

For the Bag of Words approach the results shown in figures 4-13, 4-14 and 4-15 were obtained:

accuracy 0.8357289527720739

	precision	recall	f1-score	support
usabilidad	0.69	0.72	0.71	115
sin comentario del cliente	0.50	0.59	0.54	179
quejas generales sobre vodafone	0.82	0.72	0.77	39
mal funcionamiento: general	0.42	0.67	0.52	21
mal funcionamiento: registro y logado	0.46	0.48	0.47	93
mal funcionamiento: factura	0.92	0.77	0.84	1107
nueva funcionalidad demandada	0.90	0.99	0.94	1282
mal funcionamiento: lentitud	0.53	0.60	0.56	86
micro avg	0.84	0.84	0.84	2922
macro avg	0.66	0.69	0.67	2922
weighted avg	0.85	0.84	0.84	2922

Figure 4-13: Logistic Regression – Bag of Words - Words

accuracy 0.6967830253251198

	precision	recall	f1-score	support
usabilidad	0.75	0.57	0.64	115
sin comentario del cliente	0.47	0.36	0.41	179
quejas generales sobre vodafone	0.83	0.13	0.22	39
mal funcionamiento: general	0.22	0.29	0.25	21
mal funcionamiento: registro y logado	0.49	0.37	0.42	93
mal funcionamiento: factura	0.87	0.52	0.65	1107
nueva funcionalidad demandada	0.67	0.99	0.80	1282
mal funcionamiento: lentitud	0.50	0.29	0.37	86
micro avg	0.70	0.70	0.70	2922
macro avg	0.60	0.44	0.47	2922
weighted avg	0.72	0.70	0.67	2922

Figure 4-14: Logistic Regression – Bag of Words - NGrams

accuracy 0.8579739904175222

	precision	recall	f1-score	support
usabilidad	0.75	0.74	0.74	115
sin comentario del cliente	0.49	0.60	0.54	179
quejas generales sobre vodafone	0.86	0.64	0.74	39
mal funcionamiento: general	0.57	0.76	0.65	21
mal funcionamiento: registro y logado	0.53	0.48	0.51	93
mal funcionamiento: factura	0.91	0.82	0.86	1107
nueva funcionalidad demandada	0.95	0.98	0.96	1282
mal funcionamiento: lentitud	0.52	0.72	0.60	86
micro avg	0.86	0.86	0.86	2922
macro avg	0.70	0.72	0.70	2922
weighted avg	0.87	0.86	0.86	2922

Figure 4-15: Logistic Regression – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-16, 4-17 and 4-18 were obtained:

accuracy 0.8288843258042436

	precision	recall	f1-score	support
usabilidad	0.71	0.70	0.70	115
sin comentario del cliente	0.48	0.55	0.51	179
quejas generales sobre vodafone	0.89	0.64	0.75	39
mal funcionamiento: general	0.45	0.67	0.54	21
mal funcionamiento: registro y logado	0.45	0.46	0.46	93
mal funcionamiento: factura	0.92	0.76	0.84	1107
nueva funcionalidad demandada	0.89	0.99	0.94	1282
mal funcionamiento: lentitud	0.46	0.57	0.51	86
micro avg	0.83	0.83	0.83	2922
macro avg	0.66	0.67	0.65	2922
weighted avg	0.84	0.83	0.83	2922

Figure 4-16: Logistic Regression – TF IDF - Words

accuracy 0.7070499657768652

	precision	recall	f1-score	support
usabilidad	0.70	0.62	0.66	115
sin comentario del cliente	0.44	0.35	0.39	179
quejas generales sobre vodafone	0.75	0.23	0.35	39
mal funcionamiento: general	0.24	0.29	0.26	21
mal funcionamiento: registro y logado	0.46	0.34	0.39	93
mal funcionamiento: factura	0.86	0.54	0.66	1107
nueva funcionalidad demandada	0.69	0.98	0.81	1282
mal funcionamiento: lentitud	0.46	0.30	0.37	86
micro avg	0.71	0.71	0.71	2922
macro avg	0.58	0.46	0.49	2922
weighted avg	0.72	0.71	0.69	2922

Figure 4-17: Logistic Regression – TF IDF - NGrams

accuracy 0.8528405201916496

	precision	recall	f1-score	support
usabilidad	0.72	0.75	0.73	115
sin comentario del cliente	0.46	0.59	0.52	179
quejas generales sobre vodafone	0.86	0.49	0.62	39
mal funcionamiento: general	0.61	0.67	0.64	21
mal funcionamiento: registro y logado	0.51	0.49	0.50	93
mal funcionamiento: factura	0.90	0.82	0.86	1107
nueva funcionalidad demandada	0.96	0.98	0.97	1282
mal funcionamiento: lentitud	0.45	0.60	0.52	86
micro avg	0.85	0.85	0.85	2922
macro avg	0.68	0.67	0.67	2922
weighted avg	0.86	0.85	0.86	2922

Figure 4-18: Logistic Regression – TF IDF - Characters

Looking at these results we can see that for the Logistic Regression algorithm, considering accuracy, the best approach is Bag of Words using 2-3 characters to count their frequencies, reaching 85.8% of accuracy. This is also the case when looking at the whole classification report as we can see that all categories get precisions of 49% or more.

4.1.3 Random Forest

For this algorithm I also used all the columns mentioned in chapter 3 and processed the text columns using Bag if Words and TF-IDF.

For the Bag of Words approach the results shown in figures 4-19, 4-20 and 4-21 were obtained:

accuracy 0.7628336755646817

	precision	recall	f1-score	support
usabilidad	0.54	0.69	0.61	115
sin comentario del cliente	0.47	0.41	0.44	179
quejas generales sobre vodafone	0.85	0.28	0.42	39
mal funcionamiento: general	0.20	0.24	0.22	21
mal funcionamiento: registro y logado	0.67	0.34	0.45	93
mal funcionamiento: factura	0.89	0.65	0.75	1107
nueva funcionalidad demandada	0.76	0.99	0.86	1282
mal funcionamiento: lentitud	0.58	0.34	0.43	86
micro avg	0.76	0.76	0.76	2922
macro avg	0.62	0.49	0.52	2922
weighted avg	0.77	0.76	0.75	2922

Figure 4-19: Random Forest – Bag of Words - Words

```
accuracy 0.5900068446269678
                                     precision    recall  f1-score   support

                         usabilidad       0.72      0.37      0.49       115
           sin comentario del cliente       0.65      0.19      0.29       179
      quejas generales sobre vodafone       0.40      0.05      0.09        39
          mal funcionamiento: general       0.45      0.24      0.31        21
mal funcionamiento: registro y logado       0.64      0.17      0.27        93
          mal funcionamiento: factura       0.82      0.31      0.45      1107
        nueva funcionalidad demandada       0.54      0.99      0.70      1282
         mal funcionamiento: lentitud       0.75      0.14      0.24        86

                            micro avg       0.59      0.59      0.59      2922
                            macro avg       0.62      0.31      0.36      2922
                         weighted avg       0.67      0.59      0.53      2922
```

Figure 4-20: Random Forest – Bag of Words - NGrams

```
accuracy 0.8374401095140315
                                     precision    recall  f1-score   support

                         usabilidad       0.63      0.67      0.65       115
           sin comentario del cliente       0.46      0.42      0.44       179
      quejas generales sobre vodafone       0.87      0.33      0.48        39
          mal funcionamiento: general       0.48      0.52      0.50        21
mal funcionamiento: registro y logado       0.68      0.27      0.38        93
          mal funcionamiento: factura       0.80      0.88      0.84      1107
        nueva funcionalidad demandada       0.95      0.97      0.96      1282
         mal funcionamiento: lentitud       0.68      0.35      0.46        86

                            micro avg       0.84      0.84      0.84      2922
                            macro avg       0.69      0.55      0.59      2922
                         weighted avg       0.83      0.84      0.83      2922
```

Figure 4-21: Random Forest – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-22, 4-23 and 4-24 were obtained:

```
accuracy 0.7587268993839835
                                     precision    recall  f1-score   support

                         usabilidad       0.56      0.71      0.63       115
           sin comentario del cliente       0.46      0.41      0.43       179
      quejas generales sobre vodafone       0.92      0.28      0.43        39
          mal funcionamiento: general       0.41      0.43      0.42        21
mal funcionamiento: registro y logado       0.54      0.29      0.38        93
          mal funcionamiento: factura       0.88      0.64      0.75      1107
        nueva funcionalidad demandada       0.76      0.99      0.86      1282
         mal funcionamiento: lentitud       0.67      0.34      0.45        86

                            micro avg       0.76      0.76      0.76      2922
                            macro avg       0.65      0.51      0.54      2922
                         weighted avg       0.77      0.76      0.74      2922
```

Figure 4-22: Random Forest – TF IDF - Words

accuracy 0.5937713894592744

	precision	recall	f1-score	support
usabilidad	0.79	0.42	0.55	115
sin comentario del cliente	0.57	0.22	0.32	179
quejas generales sobre vodafone	0.50	0.08	0.13	39
mal funcionamiento: general	0.56	0.24	0.33	21
mal funcionamiento: registro y logado	0.60	0.16	0.25	93
mal funcionamiento: factura	0.83	0.32	0.46	1107
nueva funcionalidad demandada	0.55	0.99	0.70	1282
mal funcionamiento: lentitud	0.79	0.13	0.22	86
micro avg	0.59	0.59	0.59	2922
macro avg	0.65	0.32	0.37	2922
weighted avg	0.67	0.59	0.54	2922

Figure 4-23: Random Forest – TF IDF - NGrams

accuracy 0.8446269678302533

	precision	recall	f1-score	support
usabilidad	0.66	0.71	0.69	115
sin comentario del cliente	0.48	0.40	0.44	179
quejas generales sobre vodafone	0.82	0.23	0.36	39
mal funcionamiento: general	0.44	0.52	0.48	21
mal funcionamiento: registro y logado	0.64	0.31	0.42	93
mal funcionamiento: factura	0.81	0.89	0.85	1107
nueva funcionalidad demandada	0.96	0.97	0.97	1282
mal funcionamiento: lentitud	0.73	0.38	0.50	86
micro avg	0.84	0.84	0.84	2922
macro avg	0.69	0.55	0.59	2922
weighted avg	0.84	0.84	0.83	2922

Figure 4-24: Random Forest – TF IDF - Characters

Looking at these results we can see that for the Random Forest algorithm, considering accuracy, the best approach is TF-IDF using 2-3 characters to count their frequencies, obtaining an accuracy of 0.845 whereas others only get 0.763 (BoW with words) or even 0.593 (IF-IDF with ngrams).This is also the case when looking at the whole classification report as we can see all categories get precisions of 44% or more.

4.1.4 Extreme Gradient Boosting

For this algorithm I used all the columns mentioned in chapter 3 as with the previous algorithms and processed the text columns using Bag if Words and TF-IDF.

For the Bag of Words approach the results shown in figures 4-25, 4-26 and 4-27 were obtained:

accuracy 0.7245037645448323

	precision	recall	f1-score	support
usabilidad	0.60	0.76	0.67	115
sin comentario del cliente	0.53	0.22	0.31	179
quejas generales sobre vodafone	0.75	0.62	0.68	39
mal funcionamiento: general	0.38	0.52	0.44	21
mal funcionamiento: registro y logado	0.71	0.22	0.33	93
mal funcionamiento: factura	0.88	0.58	0.70	1107
nueva funcionalidad demandada	0.69	0.99	0.81	1282
mal funcionamiento: lentitud	0.56	0.26	0.35	86
micro avg	0.72	0.72	0.72	2922
macro avg	0.64	0.52	0.54	2922
weighted avg	0.74	0.72	0.70	2922

Figure 4-25: Extreme Gradient Boosting – Bag of Words - Words

accuracy 0.6078028747433265

	precision	recall	f1-score	support
usabilidad	0.82	0.37	0.51	115
sin comentario del cliente	0.67	0.07	0.12	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.57	0.19	0.29	21
mal funcionamiento: registro y logado	0.61	0.15	0.24	93
mal funcionamiento: factura	0.68	0.45	0.54	1107
nueva funcionalidad demandada	0.58	0.94	0.72	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.61	0.61	0.61	2922
macro avg	0.49	0.27	0.30	2922
weighted avg	0.61	0.61	0.56	2922

Figure 4-26: Extreme Gradient Boosting – Bag of Words - NGrams

accuracy 0.8436002737850787

	precision	recall	f1-score	support
usabilidad	0.67	0.74	0.71	115
sin comentario del cliente	0.56	0.41	0.47	179
quejas generales sobre vodafone	0.85	0.72	0.78	39
mal funcionamiento: general	0.43	0.57	0.49	21
mal funcionamiento: registro y logado	0.64	0.30	0.41	93
mal funcionamiento: factura	0.83	0.85	0.84	1107
nueva funcionalidad demandada	0.92	0.98	0.95	1282
mal funcionamiento: lentitud	0.64	0.44	0.52	86
micro avg	0.84	0.84	0.84	2922
macro avg	0.69	0.63	0.65	2922
weighted avg	0.83	0.84	0.83	2922

Figure 4-27: Extreme Gradient Boosting – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-28, 4-29 and 4-30 were obtained:

accuracy 0.7323750855578371

	precision	recall	f1-score	support
usabilidad	0.60	0.78	0.68	115
sin comentario del cliente	0.65	0.28	0.40	179
quejas generales sobre vodafone	0.76	0.72	0.74	39
mal funcionamiento: general	0.52	0.52	0.52	21
mal funcionamiento: registro y logado	0.83	0.26	0.39	93
mal funcionamiento: factura	0.88	0.57	0.69	1107
nueva funcionalidad demandada	0.69	0.99	0.82	1282
mal funcionamiento: lentitud	0.58	0.34	0.43	86
micro avg	0.73	0.73	0.73	2922
macro avg	0.69	0.56	0.58	2922
weighted avg	0.76	0.73	0.71	2922

Figure 4-28: Extreme Gradient Boosting – TF IDF - Words

accuracy 0.606091718001369

	precision	recall	f1-score	support
usabilidad	0.81	0.33	0.47	115
sin comentario del cliente	0.55	0.06	0.11	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.50	0.14	0.22	21
mal funcionamiento: registro y logado	0.56	0.15	0.24	93
mal funcionamiento: factura	0.68	0.45	0.54	1107
nueva funcionalidad demandada	0.58	0.94	0.72	1282
mal funcionamiento: lentitud	0.00	0.00	0.00	86
micro avg	0.61	0.61	0.61	2922
macro avg	0.46	0.26	0.29	2922
weighted avg	0.60	0.61	0.55	2922

Figure 4-29: Extreme Gradient Boosting – TF IDF - NGrams

accuracy 0.8470225872689938

	precision	recall	f1-score	support
usabilidad	0.69	0.72	0.70	115
sin comentario del cliente	0.55	0.40	0.46	179
quejas generales sobre vodafone	0.88	0.74	0.81	39
mal funcionamiento: general	0.52	0.71	0.60	21
mal funcionamiento: registro y logado	0.69	0.31	0.43	93
mal funcionamiento: factura	0.83	0.86	0.85	1107
nueva funcionalidad demandada	0.93	0.98	0.95	1282
mal funcionamiento: lentitud	0.63	0.47	0.54	86
micro avg	0.85	0.85	0.85	2922
macro avg	0.71	0.65	0.67	2922
weighted avg	0.84	0.85	0.84	2922

Figure 4-30: Extreme Gradient Boosting – TF IDF - Characters

Looking at these results we can see that for the Extreme Gradient Boosting algorithm, considering accuracy, the best approach is TF-IDF using 2-3 characters to count their frequencies with 84.7% of accuracy against the 60.6% accuracy obtained using TF-IDF with ngrams. This is the worst accuracy obtained by this algorithm which shows that this algorithm works well with the data because it is still a good result. This is also the case when looking at the whole classification report as we can see all categories get precisions of 52% or more.

4.1.5 Stochastic Gradient Descent

For this algorithm I used all the columns mentioned in chapter 3 and processed the text columns using Bag if Words and TF-IDF.

For the Bag of Words approach the results shown in figures 4-31, 4-32 and 4-33 were obtained:

accuracy 0.7915811088295688

	precision	recall	f1-score	support
usabilidad	0.68	0.77	0.72	115
sin comentario del cliente	0.52	0.48	0.50	179
quejas generales sobre vodafone	0.88	0.56	0.69	39
mal funcionamiento: general	0.40	0.67	0.50	21
mal funcionamiento: registro y logado	0.56	0.40	0.47	93
mal funcionamiento: factura	0.93	0.68	0.79	1107
nueva funcionalidad demandada	0.79	0.99	0.88	1282
mal funcionamiento: lentitud	0.58	0.50	0.54	86
micro avg	0.79	0.79	0.79	2922
macro avg	0.67	0.63	0.63	2922
weighted avg	0.81	0.79	0.78	2922

Figure 4-31: Stochastic Gradient Descent – Bag of Words - Words

accuracy 0.5893223819301848

	precision	recall	f1-score	support
usabilidad	0.78	0.47	0.59	115
sin comentario del cliente	0.66	0.21	0.32	179
quejas generales sobre vodafone	0.00	0.00	0.00	39
mal funcionamiento: general	0.45	0.24	0.31	21
mal funcionamiento: registro y logado	0.70	0.15	0.25	93
mal funcionamiento: factura	0.89	0.30	0.45	1107
nueva funcionalidad demandada	0.53	0.99	0.69	1282
mal funcionamiento: lentitud	0.56	0.06	0.11	86
micro avg	0.59	0.59	0.59	2922
macro avg	0.57	0.30	0.34	2922
weighted avg	0.69	0.59	0.53	2922

Figure 4-32: Stochastic Gradient Descent – Bag of Words - NGrams

accuracy 0.8323066392881588

	precision	recall	f1-score	support
usabilidad	0.63	0.77	0.69	115
sin comentario del cliente	0.44	0.70	0.54	179
quejas generales sobre vodafone	0.85	0.56	0.68	39
mal funcionamiento: general	0.54	0.62	0.58	21
mal funcionamiento: registro y logado	0.59	0.41	0.48	93
mal funcionamiento: factura	0.93	0.74	0.83	1107
nueva funcionalidad demandada	0.94	0.98	0.96	1282
mal funcionamiento: lentitud	0.42	0.77	0.54	86
micro avg	0.83	0.83	0.83	2922
macro avg	0.67	0.69	0.66	2922
weighted avg	0.86	0.83	0.84	2922

Figure 4-33: Stochastic Gradient Descent – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-34, 4-35 and 4-36 were obtained:

accuracy 0.756331279945243

	precision	recall	f1-score	support
usabilidad	0.71	0.63	0.66	115
sin comentario del cliente	0.69	0.34	0.45	179
quejas generales sobre vodafone	0.89	0.44	0.59	39
mal funcionamiento: general	0.53	0.48	0.50	21
mal funcionamiento: registro y logado	0.76	0.24	0.36	93
mal funcionamiento: factura	0.88	0.67	0.76	1107
nueva funcionalidad demandada	0.71	0.99	0.82	1282
mal funcionamiento: lentitud	0.74	0.27	0.39	86
micro avg	0.76	0.76	0.76	2922
macro avg	0.74	0.50	0.57	2922
weighted avg	0.77	0.76	0.74	2922

Figure 4-34: Stochastic Gradient Descent – TF IDF - Words

accuracy 0.578713210130048

	precision	recall	f1-score	support
usabilidad	0.73	0.35	0.47	115
sin comentario del cliente	0.66	0.13	0.21	179
quejas generales sobre vodafone	1.00	0.03	0.05	39
mal funcionamiento: general	0.33	0.19	0.24	21
mal funcionamiento: registro y logado	0.61	0.12	0.20	93
mal funcionamiento: factura	0.82	0.31	0.45	1107
nueva funcionalidad demandada	0.53	0.99	0.69	1282
mal funcionamiento: lentitud	0.83	0.06	0.11	86
micro avg	0.58	0.58	0.58	2922
macro avg	0.69	0.27	0.30	2922
weighted avg	0.67	0.58	0.52	2922

Figure 4-35: Stochastic Gradient Descent – TF IDF - NGrams

accuracy 0.8494182067077344

	precision	recall	f1-score	support
usabilidad	0.71	0.76	0.73	115
sin comentario del cliente	0.65	0.38	0.48	179
quejas generales sobre vodafone	0.80	0.31	0.44	39
mal funcionamiento: general	0.39	0.71	0.51	21
mal funcionamiento: registro y logado	0.62	0.26	0.36	93
mal funcionamiento: factura	0.83	0.89	0.86	1107
nueva funcionalidad demandada	0.92	0.98	0.95	1282
mal funcionamiento: lentitud	0.76	0.36	0.49	86
micro avg	0.85	0.85	0.85	2922
macro avg	0.71	0.58	0.60	2922
weighted avg	0.84	0.85	0.84	2922

Figure 4-36: Stochastic Gradient Descent – TF IDF - Characters

Looking at these results we can see that for the Stochastic Gradient Descent algorithm, considering accuracy, the best approach is TF-IDF using 2-3 characters to count their frequencies obtaining an accuracy of 84.9% compared to the 58.9% from BoW with ngrams and the 75.6% from TF-IDF with words. This is also the case when looking at the whole classification report as we can see all categories get precisions of 39% or more.

4.1.6 Decision Tree

For this algorithm I used all the columns mentioned in chapter 3 and processed the text columns using Bag of Words and TF-IDF.

For the Bag of Words approach the results shown in figures 4-37, 4-38 and 4-39 were obtained:

accuracy 0.7618069815195072

	precision	recall	f1-score	support
usabilidad	0.58	0.65	0.61	115
sin comentario del cliente	0.40	0.42	0.41	179
quejas generales sobre vodafone	0.72	0.33	0.46	39
mal funcionamiento: general	0.26	0.29	0.27	21
mal funcionamiento: registro y logado	0.36	0.37	0.36	93
mal funcionamiento: factura	0.88	0.65	0.75	1107
nueva funcionalidad demandada	0.81	0.99	0.89	1282
mal funcionamiento: lentitud	0.41	0.42	0.41	86
micro avg	0.76	0.76	0.76	2922
macro avg	0.55	0.51	0.52	2922
weighted avg	0.77	0.76	0.75	2922

Figure 4-37: Decision Tree – Bag of Words - Words

accuracy 0.5982203969883642

	precision	recall	f1-score	support
usabilidad	0.69	0.46	0.55	115
sin comentario del cliente	0.61	0.22	0.32	179
quejas generales sobre vodafone	0.43	0.08	0.13	39
mal funcionamiento: general	0.45	0.24	0.31	21
mal funcionamiento: registro y logado	0.48	0.24	0.32	93
mal funcionamiento: factura	0.82	0.31	0.46	1107
nueva funcionalidad demandada	0.56	0.99	0.71	1282
mal funcionamiento: lentitud	0.70	0.16	0.26	86
micro avg	0.60	0.60	0.60	2922
macro avg	0.59	0.34	0.38	2922
weighted avg	0.67	0.60	0.55	2922

Figure 4-38: Decision Tree – Bag of Words - NGrams

accuracy 0.768309377138946

	precision	recall	f1-score	support
usabilidad	0.57	0.63	0.60	115
sin comentario del cliente	0.27	0.31	0.28	179
quejas generales sobre vodafone	0.69	0.62	0.65	39
mal funcionamiento: general	0.18	0.43	0.25	21
mal funcionamiento: registro y logado	0.26	0.31	0.28	93
mal funcionamiento: factura	0.82	0.69	0.75	1107
nueva funcionalidad demandada	0.93	0.98	0.95	1282
mal funcionamiento: lentitud	0.34	0.44	0.39	86
micro avg	0.77	0.77	0.77	2922
macro avg	0.51	0.55	0.52	2922
weighted avg	0.79	0.77	0.77	2922

Figure 4-39: Decision Tree – Bag of Words – Characters

For the TF-IDF approach the results shown in figures 4-40, 4-41 and 4-42 were obtained:

accuracy 0.7638603696098563

	precision	recall	f1-score	support
usabilidad	0.65	0.71	0.68	115
sin comentario del cliente	0.44	0.46	0.45	179
quejas generales sobre vodafone	0.54	0.33	0.41	39
mal funcionamiento: general	0.16	0.24	0.19	21
mal funcionamiento: registro y logado	0.37	0.34	0.36	93
mal funcionamiento: factura	0.87	0.65	0.75	1107
nueva funcionalidad demandada	0.81	0.99	0.89	1282
mal funcionamiento: lentitud	0.40	0.33	0.36	86
micro avg	0.76	0.76	0.76	2922
macro avg	0.53	0.51	0.51	2922
weighted avg	0.77	0.76	0.76	2922

Figure 4-40: Decision Tree – TF IDF - Words

accuracy 0.5999315537303217

	precision	recall	f1-score	support
usabilidad	0.73	0.50	0.59	115
sin comentario del cliente	0.57	0.21	0.30	179
quejas generales sobre vodafone	0.43	0.08	0.13	39
mal funcionamiento: general	0.33	0.24	0.28	21
mal funcionamiento: registro y logado	0.54	0.20	0.30	93
mal funcionamiento: factura	0.83	0.32	0.46	1107
nueva funcionalidad demandada	0.56	0.99	0.71	1282
mal funcionamiento: lentitud	0.70	0.16	0.26	86
micro avg	0.60	0.60	0.60	2922
macro avg	0.59	0.34	0.38	2922
weighted avg	0.67	0.60	0.55	2922

Figure 4-41: Decision Tree – TF IDF - NGrams

accuracy 0.7864476386036962

	precision	recall	f1-score	support
usabilidad	0.51	0.63	0.57	115
sin comentario del cliente	0.37	0.45	0.40	179
quejas generales sobre vodafone	0.62	0.51	0.56	39
mal funcionamiento: general	0.14	0.38	0.20	21
mal funcionamiento: registro y logado	0.33	0.41	0.36	93
mal funcionamiento: factura	0.85	0.72	0.78	1107
nueva funcionalidad demandada	0.95	0.97	0.96	1282
mal funcionamiento: lentitud	0.37	0.44	0.40	86
micro avg	0.79	0.79	0.79	2922
macro avg	0.52	0.56	0.53	2922
weighted avg	0.81	0.79	0.80	2922

Figure 4-42: Decision Tree – TF IDF - Characters

Looking at these results we can see that for the Decision Tree algorithm, considering accuracy, the best approach is TF-IDF using 2-3 characters to count their frequencies obtaining an accuracy of 0.786 against the accuracies obtained with BoW with ngrams (0.598) or the one obtained with BoW with characters (0.768). This is also the case when looking at the whole classification report as we can see all categories get precisions of 14% or more.

After seeing all these results, one might ask herself: "why most of the best results are obtained when counting characters' frequencies instead of words or even ngrams?". I believe these results are obtained because the comments are not very long, some of them are only one or two words, and that makes it harder for the algorithm to find differences between categories when counting word and ngram frequencies. This is why I think with 2-3 characters' frequencies the algorithms are able to find some patterns for each category and are able to predict them better.

4.1.7 Convolutional Neural Network

For this algorithm I used the Verbatim and Categories columns, and Word2Vec tokenization for text processing. I used the layers shown in figure 4-43 to create this network. For the configuration of this algorithm I found some examples and all of them had the following layers, changing the number of convolutional and maxpooling layers. I decided to add all 3 pairs to try and obtain a better result.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 1000)              0
_________________________________________________________________
embedding_1 (Embedding)      (None, 1000, 100)         501300
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 996, 128)          64128
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 199, 128)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 195, 128)          82048
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 39, 128)           0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 35, 128)           82048
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 1, 128)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 1032
=================================================================
Total params: 747,068
Trainable params: 747,068
Non-trainable params: 0
_________________________________________________________________
```

Figure 4-43: Convolutional Neural Network Layers

With this approach the result shown in figures 4-44 was obtained:

```
Loss: 1.491
Accuracy: 0.439
```

Figure 4-44: Convolutional Neural Network Results

4.1.8 Recurrent Neural Network

For this algorithm I used the same approach as in the previous algorithm: Verbatim and Categories columns and word2vec for text processing. I used the layers shown in figure 4-45 to create this network. For the configuration of this algorithm I found that the best results were obtained using the bidirectional Long Short-Term Memory since it not only uses the information from the past state but also from the future one. This method increases the amount of information available and I believed it would help obtain better results.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 1000)              0
_________________________________________________________________
embedding_1 (Embedding)      (None, 1000, 100)         629500
_________________________________________________________________
bidirectional_1 (Bidirection (None, 200)               160800
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 1608
=================================================================
Total params: 791,908
Trainable params: 791,908
Non-trainable params: 0
_________________________________________________________________
```

Figure 4-45: Recurrent Neural Network Layers

With this approach the result shown in figures 4-46 was obtained:

```
Loss: 1.423
Accuracy: 0.602
```

Figure 4-46: Recurrent Neural Network Results

4.1.9 Comparative

Figure 4-47 shows the best results found for each algorithm tested and explained previously:

Algorithm	Accuracy	Precision (lowest value)	Precision (highest value)
Naïve Bayes	0.798	0%	100%
Logistic Regression	0.858	49%	95%
Random Forest	0.845	44%	96%
Extreme Gradient Boosting	0.847	52%	93%
Stochastic Gradient Descent	0.849	39%	92%
Decision Tree	0.786	14%	95%
Convolutional Neural Network	0.439	-	-
Recurrent Neural Network	0.602	-	-

Figure 4-47: Recurrent Neural Network Results

Seeing these results, we can conclude that the best method for classifying the texts written by the clients included in our dataset is Extreme Gradient Boosting since it has one of the highest accuracies (84.7%), has the highest precision (52% in the worst case) and all the categories are predicted. Therefore, this will be the algorithm used for the solution created for the company.

5 TECHNOLOGY USED AND INTEGRATION

5.1 TECHNOLOGIES USED

This project has been developed in Python [37], since it is a very powerful programming language but at the same time it is easy to program and understand. In addition, there are some libraries available to process text in Python such as NLTK [38] and for machine learning like Scikit-Learn [39], which I have used. Finally, Python will let me create an effective communication protocol with the software currently used in Vodafone, helping with the integration.

Jupyter Notebook [40] was used during the beginning of the project since it is a very useful application with which to try all the different algorithms and text processing methods without having to install anything locally. It also helps with data visualization and is very easy to use.

5.2 INTEGRATION

As can be seen in results described in section 4, the algorithm that provided a better overall categorization was Extreme Gradient Boosting which means that this algorithm is the one chosen to create the solution for the company.

The company asked for an easy to use solution that created a csv file with the predicted categorization as its output, given a dataset also in csv format. To create this solution, I designed and developed three python scripts:

- <u>Train</u>
  This script reads the data collected through the questionnaires for clients from a csv file in utf-8 encoding and processes it to remove non wanted columns and only keep the ones mentioned in section 3.1 as well as filtering it to work only with comments from clients who are detractors. After this initial processing all remaining columns are processed using the methods explained in section 3.2 and basic encoders for numeric columns. Finally, it trains the algorithm chosen with the processed data(fit). After the training is completed, the model is saved in a pickle file to be able to use it in the future without having to train it every time. This will be possible as long as the csv trained and the csv to categorize have the same categories and form names. In any other case, this script can be easily updated to fit different requirements.

- <u>Categorize</u>
  This script reads the data collected through questionnaires from a csv file in utf-8 encoding and processes it in the same way as the previous script to have a unified input for the model; then it obtains the previously trained model and uses it to predict the categories for the new processed data. To finish, this script creates a new csv file,

copied from the one given as input, and adds a new column with the predicted categories. This new csv is saved locally under a name given in the execution of the script.

- <u>Compare</u>
  This script was created to offer the possibility of comparing the categorization output with the original categorization, provided that the dataset had previously been categorized by hand. In other words, it supports testing the automatic categorization. This must receive as input a dataset different from the one used for training, since the algorithm needs different data to train and predict to be able to obtain correct results. To do the comparison, this script uses the csv created by the categorization script, which includes both automatic and handmade categorizations.

With these scripts, the company will be able to categorize the new verbatim they obtain every month in an easy and fast way since they normally receive about 2400 free-text comments in a single month.

There are other integration possibilities, like adding, in the scripts, direct access to the company's database to read and write the data. This is mentioned in the future work section of this thesis, along with other possibilities, like integrating with Vodafone's interface application, that are being evaluating.

6 CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

Vodafone gets around 2400 comments from their clients each month and they need to categorize them in order to get information about the services, products or aspects they are commenting on. Before this thesis they did this by hand, which is why they offered a scholarship for master students who would focus their master thesis on creating a solution that could automatize this process. This is the main goal of this thesis' work and I was awarded that scholarship to work on it.

During the development of this thesis I have investigated ways to process these comments and algorithms that could learn the way they are categorized to be able to predict this new categorization. I have tried various different approaches and the best one found for our dataset was Extreme Gradient Boosting using TF-IDF text processing calculating the characters' frequencies. This approach reached one of the best accuracies, 84.7% and the best precision with all categories being predicted with more than 52% of precision.

With this approach I created Python scripts to train the model and categorize new data. They were separated so they are faster and easier to use in the company. They can also be modified easily if we want them to work with different natural language processing techniques or different machine learning. These scripts have recently been tested at the company and will soon be used to categorise new comments every month.

The results in this project are expected to have a high impact and become a real and substantial improvement for a company such as Vodafone, which, by incorporating automatic analysis of its data, will be able to exploit its resources and focus more on the development of strategies to upgrade its customer service and those specific services that are the cause of the negative comments and, in this way, be able to keep the detractors from leaving the company.

Finally, I would like to say that during the time spend with this Master's Thesis I have acquired new knowledge on natural language processing, classification algorithms and also how a big company like Vodafone works. This thesis has also helped me learn more about myself and how I can accomplish my goals, whatever they might be.

6.2 FUTURE WORK

In this thesis I have been able to meet the company's goals and good results have been obtained, but there is still a lot that can be done with NLP and classification algorithms. Concerning the classification algorithms, there are so many that I could not try that maybe one of them could give us better results so, as future work, one thing that could be done is try different algorithms to see if even better results can be obtained. Also, more testing, with more data collected from the company in the future, could be done in order to make sure the solution created is the best for the data the company has. In fact, it would be possible to configure the script to use it with different algorithms depending on which obtains the best results at each time.

Another interesting idea is doing a deeper analysis on the clients' comments to try to get their feelings (through specific sentiment analysis techniques [41]) and what the actual specific problem is (not only its category, which is now predicted thanks to this work, but also the specific issue to be addressed, if any). This would benefit the company even more, since this way they could know, automatically, not only which services or aspects are the least valued and are in need of upgrading, but also what are the main complaints about, like some specific product or service is too slow or there is a need to add "how-to" pages, and what they should fix to be able to keep these clients happy in the company.

From the implementation point of view, since the beginning of this project I considered the possibility of connecting the developed scripts with the company's database to obtain the data directly from it and add the predictions back to it. Now that the feasibility of this approach to help with the clients' valuations analysis has been demonstrated and has been materialized into a concrete solution with good results, that possibility can be evaluated at the company.

Last but not least, and also aiming to completely integrate the solution developed with the Vodafone software, we will explore the possibility of showing the results obtained, along with other data stored, in a graphical interface where one can see how many comments are in each category and the specific problems most users are complaining about. With this interface they should also be able to see a client satisfaction evolution through the year and have an overview of the clients' most and least valued services and the reason behind it, which could be obtained with the deeper analysis proposed above.

REFERENCES

[1] https://www.vodafonecampuslab.es/becas/
[2] https://dictionary.cambridge.org/dictionary/english/language
[3] Keith D. Foote, A Brief History of Natural Language Processing (NLP), DATAVERSITY, 2019 [https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/]
[4] Dr. Mihael J. Garbade, A Simple Introduction to Natural Language Processing, Becoming Human, 2018 [https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32]
[5] https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7
[6] https://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp/
[7] http://www.insightsbot.com/blog/R8fu5/bag-of-words-algorithm-in-python-introduction
[8] https://skymind.ai/wiki/bagofwords-tf-idf
[9] https://www.geeksforgeeks.org/tf-idf-model-for-page-ranking/
[10] https://www.kdnuggets.com/2018/08/wtf-tf-idf.html
[11] https://skymind.ai/wiki/word2vec
[12] https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa
[13] Bernard Marr, A Short History of Machine Learning – Every Manager Should Read, Forbes, 2016 [https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/#9597f215e780]
[14] https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248
[15] https://towardsdatascience.com/scikit-learn-decision-trees-explained-803f3812290d
[16] https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[17] https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/
[18] https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[19] https://towardsdatascience.com/random-forest-3a55c3aca46d
[20] https://medium.com/datadriveninvestor/using-extreme-gradient-boosted-trees-in-machine-learning-classification-problems-a7bb04be759
[21] https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
[22] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, “Recurrent Convolutional Neural Networks for Text Classification”, AAAI Publications, Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015
[23] Yoon Kim, “Convolutional Neural Networks for Sentence Classification”, Cornell University, 2014 [https://arxiv.org/abs/1408.5882]
[24] Ji Young Lee, Franck Dernoncourt, “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks”, Cornell University, 2016 [https://arxiv.org/abs/1603.03827]

[25] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[26] https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce
[27] S. Sun, C. Luo, J. Chen, “A review of natural language processing techniques for opinion mining systems”, Information Fusion, Vol.36, July 2017, pp. 10-25 [doi:10.1016/j.inffus.2016.10.004]
[28] P. Rodriguez, A. Ortigosa, R. M. Carro, “Detecting and making use of emotions to enhance student motivation in e-learning environments”, International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL), Vol.24 (2), 2014, pp.168-183 [doi: 10.1504/IJCEELL.2014.060156]
[29] M. Fernández-Gavilanes, T. Álvarez-López, J. Juncal-Martínez, E. Costa-Montenegro, F.J. González-Castaño, “Unsupervised method for sentiment analysis in online texts”, Expert Systems with Applications Vol. 58, 2016, pp. 57-75 [doi: 10.1016/j.eswa.2016.03.031]
[30] A. Ortigosa, J.M. Martín, R.M. Carro, “Sentiment analysis in Facebook and its application to e-learning”, Computers in Human Behavior Vol. 31, Elsevier Science Publishers, 2014, pp. 527-541 [doi: 10.1016/j.chb.2013.05.024]
[31] M. Hu, B. Liu, “Mining Opinion Features in Customer Reviews”, Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 22-25, 2004 Seattle, WA, USA [doi: 10.1145/1014052.1014073]
[32] R. Piryani a, D. Madhavi b, V.K. Singh, “Analytical mapping of opinion mining and sentiment analysis research during 2000–2015”, Information Processing and Management Vol. 53, 2017, pp. 122–150 [doi: /10.1016/j.ipm.2016.07.001]
[33] https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568
[34] https://www.nltk.org/book/ch06.html
[35] https://stackabuse.com/text-classification-with-python-and-scikit-learn/
[36] https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
[37] https://www.python.org/
[38] NLTK, http://www.nltk.org/yhttps://www.nltk.org/book/ch01.html
[39] Scikit Learn, https://scikit-learn.org/stable/
[40] https://jupyter.org/
[41] M.D. Devika, C. Sunitha, Amal Ganesh “Sentiment Analysis: A Comparative Study on Different Approaches”, Department of CSE, Vidya Academy of Science and Technology, Thrissur 680501, India, 2016 [https://doi.org/10.1016/j.procs.2016.05.124]
[42] T. K. Das, D. P. Acharjya and M. R. Patra, "Opinion mining about a product by analyzing public tweets in Twitter," 2014 International Conference on Computer Communication and Informatics, Coimbatore, 2014, pp. 1-4. [doi: 10.1109/ICCCI.2014.6921727]
[43] P. V. Rajeev and V. S. Rekha, "Recommending products to customers using opinion mining of online product reviews and features," 2015 International Conference on

Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, 2015, pp. 1-5 [doi: 10.1109/ICCPCT.2015.7159433]

[44] Necmiye Genc-Nayebi, Alain Abran, “A systematic literature review: Opinion mining studies from mobile app store user reviews”, Department of Software Engineering and Information Technologies, École de technologie supérieure, University of Québec, 1100 Rue Notre-Dame O, Montreal, QC H3C 1K3, Montréal, Canada, 2016 [https://doi.org/10.1016/j.jss.2016.11.027]

[45] https://towardsdatascience.com/from-word-embeddings-to-pretrained-language-models-a-new-age-in-nlp-part-1-7ed0c7f3dfc5

GLOSSARY

NLP	Natural Language Processing.
Verbatim	Users’ comments and opinions in text form.
NPS	Net Promoter Score, the difference between the percentage of promoters and detractors.
Detractor	Client who entered a 6 or less in the recommendation field of the questionnaire
BoW	Bag of Words