University of Valladolid

COMPUTER SCIENCE FACULTY OF SEGOVIA

Bachelor in Computer Science

Product brand analysis using NLP techniques

Student: Alejandro Barrio Mateos

Mentors: José Vicente Álvarez Bravo
Silvia Duque Moro

Date: 30 June 2023

Product brand analysis using NLP techniques

Alejandro Barrio Mateos

30 June 2023

Contents

List of Figures

List of Tables

Index of code

To my grandfather Abelardo,
for having been my sidekick for 22 wonderful years;
to my parents, José Antonio and Ascensión,
because all I am today is thanks to you;
and to all my friends,
for being there whenever I need you.

Abstract

Nowadays, social media have a great influence on both society and business, as they allow brands to know what their customers think about their products and services, giving them the opportunity to improve their marketing and communication strategies, among others. In this way, social media have become an important source of information and analysis.

In this context, the aim of this project is to carry out an analysis of the information published on Twitter associated with a subset of brands, making a comparison of their presence and impact on social networks. To do this, the focus will be on the analysis of textual information collected both by social networks and through product review pages, making a study of the topics to be addressed as well as the sentiment associated with these texts.

Keywords: social media analytics, web scraping, Twitter, natural language processing, topic modelling, sentiment analysis.

Part I

Description of the project

Chapter 1

Introduction

The Internet has become an everyday tool for everyone. We spend most of our lives online: from finding out how to make dinner, to commenting on the latest episode of our favourite series. But are we really aware of the amount of data we generate on a daily basis? In 2021, there were an estimated 4.66 billion active internet users worldwide, generating more than one trillion megabytes of data every day, which means that approximately 60% of the world's population is online [7]. As shown in the figure 1.1, It is estimated that by 2024 the amount of data on the Internet will reach 149 zettabytes.

Much of this information comes from what is known as social media. Although at first glance may only refer to social networks, is an even larger part of the web. **Social media** are the platforms on which people interact and socialise, forming communities in which to share ideas, news or interests. In contrast to traditional media, where content is generated by a single large broadcaster, social media allows the community that consumes the content to be responsible for generating it. Interaction is a key factor in social media [29].

These features are crucial on a variety of platforms, so social media are used for a wide variety of purposes:

- Maintain microblogs and keep up to date with current events (e.g. via Twitter).
- Create and share multimedia content (e.g. via YouTube)
- Find answers to specific questions (e.g. via Stack Overflow)
- Read reviews and opinions, both from users and specialised critics, of films, series or video games (e.g. via Metacritic)
- Keep in touch with family and friends (e.g. via Facebook)
- Know the ratings of other users who have been to the same restaurant you plan to go to (e.g. via Google Maps)

Figure 1.1: Statistics of data generated daily during 2021.

1.1 Problem statement

Given the vast amount of data available to us, the question arises as to what conclusions we can draw from it. This question arises especially in the business world, as this data can be a source of real-time information on market trends and consumer behaviour, enabling management teams to make more accurate and realistic decisions. This school of thought is supported by research showing that 97.2% of companies are already investing in data analytics and artificial intelligence projects, 24% of which employ data-driven decision making. The rise of the data-driven approach is mainly due to the fact that companies using data structuring and analysis methods have seen their profits increase on average by 8% [42].

This interest is also reflected in the job market, where tens of thousands of job offers requiring data analytics skills are available all over the world [43]. Specifically, last year's report on data skills by developer interview and recruitment platform DevSkiller saw a 295% increase in the number of data science-related tasks that job recruiters were setting for candidates in the interview process during 2021. This increase in demand was already predicted by authorities such as the UK's Royal Society, which in 2019 warned that demand for data scientists and engineers had tripled in the last five years, leaving companies "desperate to find professionals to unlock the potential of new technologies" such as machine learning and artificial intelligence [32].

However, the increase in demand for such positions has not grown in line with market capabilities, but rather the opposite. This has led to what is known as the **data skills gap**, as professionals in the sector have not had the opportunity to train quickly enough to acquire the skills in demand. This has important consequences. By 2030, labour shortages in the technology, media and telecommunications sector could cause the US to lose an estimated 162 billion, according to a 2018 report by Korn Ferry. Globally, the digital skills gap could cause 14 G20 countries to lose 11.5 trillion in cumulative gross domestic product growth, according to Salesforce estimates [25].

The ultimate aim of this work is to address both challenges at the same time by generating a series of interactive Jupyter notebooks for the extraction, screening and analysis of information relating to a number of companies together with user feedback on the most recent releases of those companies. This will provide students with learning material to serve as a starting point for future workers in the field of social data analysis, as well as an effective study from which the companies involved can draw immediate conclusions about the impact of their products on the community.

1.2 Objectives of the work

The project seeks to solve the above problem by focusing on textual social data relating to a subset of companies and their most recently launched products. Specifically, it aims to achieve the following objectives:

- **OBJ-1:** Generate a set of interactive notebooks to serve as an introduction to social data mining and analysis.
- **OBJ-2:** Identification of the main themes associated with the brand through topic modelling.
- **OBJ-3:** Detection of the public opinion shown by the community regarding the most recent launches of each company.
- **OBJ-4:** Correlation of this data with sales and public reception.

1.2.1 Restrictions

However, the different constraints encountered during the development of the project must also be taken into account:

- **REST-1:** The project has been developed on a computer running Windows 10, using Python version 3.0 and pip 22.2.2, which means that some older implementations of libraries have caused problems, making it necessary to resort to more recent versions (as is the case of the change of library used to implement the biterm model).
- **REST-2:** The scope and duration of the project must be within the standards established by the Bachelor's Degree Project teaching guide (12 ETCS).
- **REST-3:** During the process of planning and documenting the work, Elon Musk bought Twitter. One of his first decisions was to turn the Twitter API into a paid service as of 9 February 2023, which precipitated the data collection process. [34]

1.3 Business context

It should be noted that this Bachelor's Degree Project is the result of a collaboration between the University of Valladolid and the company NielsenIQ, which focuses on the analysis of data related to user behaviour and consumption.

NielsenIQ was founded as part of the New York-based Dutch-American media conglomerate known as Nielsen Holdings. It is now the world's leading audience measurement, product and service data analytics and marketing information provider [1]. Combining the use of complex analytical tools and integrated marketing solutions, Nielsen offers clients a global view of both their market and their customers. With multiple locations around the world, the company operates in more than 100 countries with a global team dedicated to helping clients effectively uncover previously hidden opportunities [2].

Specifically, NielsenIQ focuses on providing the most comprehensive view of consumer behaviour on an unbiased global basis, focusing on the world's leading consumer goods companies and retailers. It does this by leveraging a comprehensive data set measuring all transactions equally, providing clients with a forward-looking view of consumer behaviour. Its open philosophy on data integration allows for a broad base on which to work and research [3].

As of March 2021, NielsenIQ was acquired by private equity investor Advent International, changing its status to that of an independent company. This paradigm shift has enabled NielsenIQ to accelerate its transformation and strengthen its position as a leader in the analytics market [47].

1.4 Structure of the memory

The work is organised according to the following outline:

- **Chapter 1.** Introduction. General description of the problem to be addressed, as well as a first outline of the business environment in which the project is framed. Delimitation of objectives, identification of restrictions and an outline of the structure of the document.
- **Chapter 2.** Planning. Work methodology to be followed, setting out the tasks to be carried out within the foreseen time frame. An economic and temporal balance is also presented, linked to the costs derived from the achievement of the project, verifying whether the reality has been adjusted to the foreseen planning.
- **Chapter 3.** Context of the work. Where the project is framed within the current reality, both in the general and specific environment, presenting the basic terminology of data science focused on natural language processing. It also presents some existing works that carry out studies similar to the one in this project.
- **Chapter 4.** Data extraction. Presentation of the two most common techniques for collecting information written on the web: the use of APIs and the technique known as web scraping. The implementation of these techniques is shown, with an analysis of the data structures obtained.
- **Chapter 5.** Data pre-processing. This section shows the most common techniques for cleaning the information obtained through the web, applying these processes to the information that has been collected in the previous section and thus having a final set of useful data.
- **Chapter 6.** Topic modeling. Throughout this section, an analysis of the different topics addressed in the texts obtained is carried out, firstly by means of the preliminary analysis offered by the technique of wordcloud, and then in depth analysis of the different topics thanks to topic detection models: LDA and BTM.
- **Chapter 7.** Sentiment analysis. Finally, one of the most powerful techniques related to the analysis of texts produced on the Internet is introduced: sentiment analysis. Pre-trained models will be applied to draw such conclusions, even presenting our own model for detecting the general sentiment of reviews.
- **Chapter 8.** Conclusions and future work. To conclude, a final comparison is made between the companies and their products based on the information gathered throughout the study. Possible improvements that could be made in future iterations of the work are also presented.
- **Appendix A.** Installation manual for the runtime environment that allows viewing and interacting with Jupyter notebooks.

- **Appendix B.** Content attached to the report.
- **Bibliography.**

Chapter 2

Planning

2.1 Methodology of work

The methodology chosen for the development of the project is Scrum [57]. Scrum is an agile project management framework that allows teams to structure and manage work through a set of values, principles and practices. Scrum stands out for allowing projects to be approached with a focus on empiricism and lean -one that reduces waste and focuses on what is essential-, being an ideal heuristic for developments such as the one proposed in this paper, in which the development team (in this case the student) lacks full experience, which will increase throughout the project thanks to continuous learning and adaptation to fluctuating factors. This adaptability of Scrum to changing needs is also ideal for project planning [21].

Scrum is based on the division of the development team into different **roles**. However, given that we are dealing with individual development, it will be necessary to adapt these to our situation:

- **Developer.** Role embodied by the student who carries out the project. His mission is to develop the Bachelor's Degree Project, both its implementation and the report that will be delivered as the final result.
- **Product owner.** Responsible for maximizing the value of the resulting product generated after each iteration, as well as giving a first idea of the Product Backlog. This role is played by the company that commissions the student to carry out the TFG, represented by the company's internship tutor.
- **Scrum Master.** Responsible for ensuring that the guidelines of the Scrum framework are followed. This role is shared both by the tutors from the university and the company, as well as by the student himself, who are ultimately responsible for the adequacy of their efforts to the selected methodology.

Apart from the roles, Scrum takes as a basis for the structuring of tasks the **events**, which provide the development with transparency in order to inspect and improve them throughout the development process. We will distinguish the following events for our project:

- **Sprint.** Each sprint can be considered as a small-scale project. During the lifetime of a sprint, changes can be made to the Product Backlog or even the scope can be redefined, as long as the commitment to quality is maintained and such changes do not jeopardise the ultimate goal of the sprint.

  For this particular project, it has been estimated that 5 sprints of 2 weeks each will be necessary [50], with a load of 60 hours per sprint.

- **Sprint planning.** First event held before starting a sprint. Meeting between the student and one of the tutors, in which the objectives to be achieved in the sprint to carry out are set, the User Stories of the Product Backlog to be developed and the plan. The summary of everything is called the Sprint Backlog. The first planning session was carried out individually, but from the second sprint onwards it was unified with the Mentored Meeting event to speed up the planning processes, due to the short duration of the sprints.

- **Daily (Scrum).** Daily meeting focused on analysing the progress made in the development and, if necessary, adapting the Sprint Backlog to the established work plan. As the work is carried out by a single person, this meeting consists of reviewing the objectives set and modifying them if necessary.

- **Mentored Meeting.** Event aimed at evaluating the progress achieved during the sprint, setting possible points of improvement for the next iteration of the delivered product. These improvements are also analyzed in these meetings to optimize time, as they are held between the student and the company tutor or the university tutor. This event unifies the Scrum concepts known as Sprint Review and Sprint Retrospective.

Lastly, we find the **artefacts**, which represent work or value. Their design is focused on transparency, in order to achieve the standard of progress that each of them models:

- **Product Backlog.** An ordered list of what needs to be improved in the final product. Its components can be refined and even broken down into more precise and tangible ones, at which point they are added to the Sprint Backlog of the planning. Its ultimate aim is to ensure that the **product objective** -a long-term goal that defines the ideal future state of the project and that must be met before a new one can be assigned- is reached.

- **Sprint Backlog.** Planning by and for developers. It consists of the sprint goal, a list of the sprint-specific elements of the Product Backlog, as well as the plan for carrying out the increment (why, what and how; respectively). It allows the developer to be clear about the sprint goal, the element against which the Sprint Backlog is committed. It is a concrete objective that is defined during planning.

- **Increment.** Specific step aimed at achieving the product goal. Each increment is added to those already achieved, also adding value by being a usable element. An increment culminates when the so-called Definition of Done (DoD) is reached, a state in which the set quality standards have been achieved.

  For the development of this project, it was common the delivery of increments to the company that provided the Bachelor's Degree Project, as it is one of the main stakeholders of the work.

2.1.1 Methodology of work in data science

The Scrum methodology is valid for the development of projects regardless of their nature. However, since this work focuses specifically on the development of a data science project, it will be convenient to study the prototypical phases that make up projects of this nature [17].

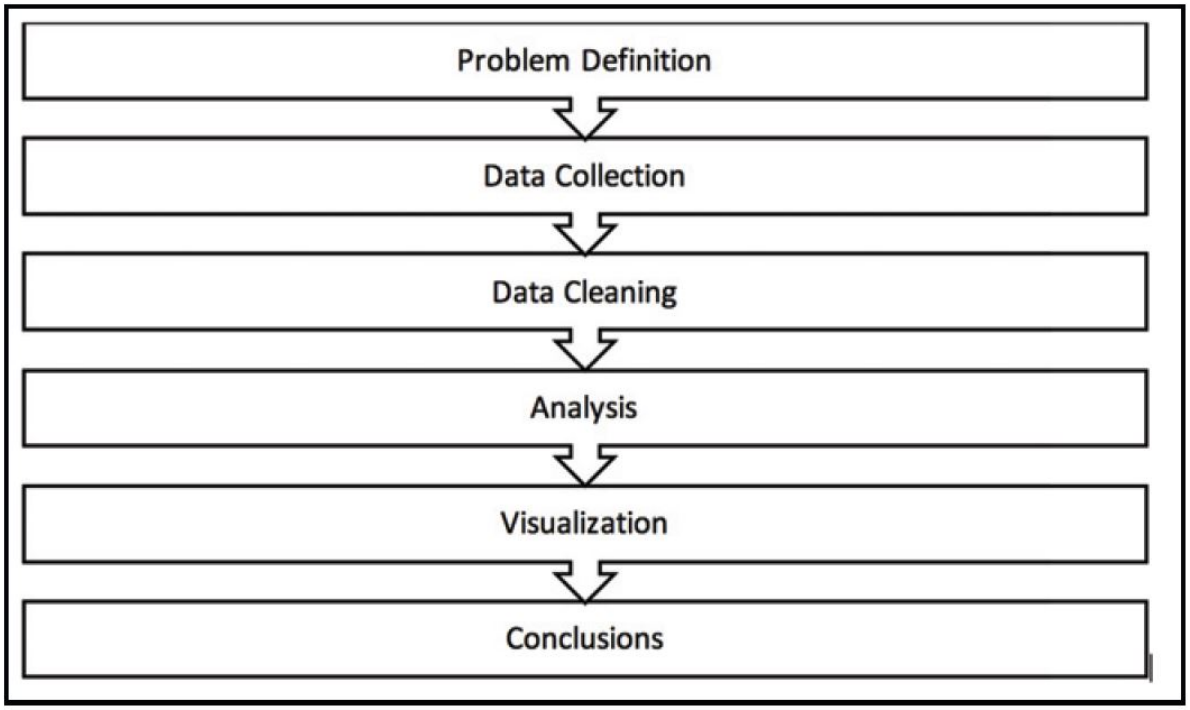

Figure 2.1: Steps for data extraction, processing and analysis

1. **Definition of the problem:** A proper understanding of the problem will allow us to choose the right sources of information, as well as the right methods to analyze the data and the conclusions we hope to draw. This is crucial, because if you do not define a clear objective when conducting a study, you may end up with contaminated data (e.g. ads generated by bots) that do not provide any relevant conclusions.

2. **Data collection:** Once the area of study on which the project will focus has been chosen, it will be necessary to acquire the data to be analyzed. Depending on the source from which the data is collected, this task can be carried out by using an API that facilitates its automated extraction (as in the case of Twitter and its API); or through a somewhat more complex procedure known as "scraping", aimed at obtaining information from media that do not provide such facilities (as in the case of blogs or forums).

3. **Data cleaning:** After storing all the data that could be considered relevant, it will be necessary to sift through it to get rid of all the data that is not of interest. Examples of this are using a language filter or discarding duplicate data. The ultimate goal of this phase is to obtain a clean dataset to start working with.

4. **Analysis:** In the analysis process, we will have to define what type of study we will conduct, as well as the best structure for our data. The choice of analysis method depends entirely on the final objective of the study, as it can vary from the use of basic statistical techniques to the need to employ Machine Learning models.

5. **Visualization of the results:** The best way to understand the results obtained during the analysis phase is usually always through graphical representations, which are enlightening both for the analyst (who may discover new features in the data that motivate a second analysis of the information from a different perspective) and for the client who commissioned the study.

6. **Conclusions:** Based on the information obtained during the analysis and visualization of the data, it is time to draw conclusions in order to finalize the study and make a final presentation of all the information obtained.

Throughout the development process, as well as in this document, an attempt has been made to structure the planning, design and implementation of the project according to the order established by this outline.

2.2 Time planning

The sprint Planning event requires an estimation phase, in which the approximate time required to achieve the different tasks that make up the sprint is calculated. There are a wide variety of methods to carry out this allocation, although the one used in this development is known as Planning Poker.

Planning Poker [4] is an estimation technique for agile projects, aimed at seeking consensus among the different parties involved in the development of the project to assess the effort required to achieve a user story. Team members will assign each story a value from the Fibonacci sequence (1, 2, 3, 5, 8...), with the value most voted by the team being chosen as the final value. It should be noted that the score obtained can never be converted into units of time, as what is represented by this value is:

- Amount of work to be done
- Story or task complexity
- Risks or uncertainties that may arise in the course of development

For this project, despite being an individual development, the use of this methodology for the definition of story points has allowed the planning and structuring of the tasks to be carried out in a systematic way, better defining the effort required for each sprint when it comes to placing it in time. When working with short sprints, each one is composed of a single user story, decomposed into several tasks. In total, we have obtained a total of 5 user stories encompassing a set of 85 story points, each having approximately 17 story points assigned to it, which would ensure a balanced distribution of work between the different sprints.

In addition, the dynamic planning of the sprints has allowed a development with a greater degree of freedom, adjusting to both the needs of the project (such as the collection of data by Twitter due to the future payment restrictions of the servers), and those of the different actors involved in the development process (such as the conflict with other responsibilities of the student - exams of subjects of the bachelor as well as another Bachelor's Degree Project and company internship - in order to be able to continue with the implementation of the work).

2.2.1 Sprint #1

During the first sprint, the execution of the tasks that make up the first user story [2.1] was carried out, focusing on the acquisition of basic notions within the context of the work and seeking to delimit the original focus of the project, as after an initial analysis it was decided that a more specific focus on objectives would allow the development of a higher quality product in the given time.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	07/11/2022	08/11/2022	09/11/2022	10/11/2022	11/11/2022	12/11/2022	13/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	19/11/2022	20/11/2022
H1	**Documentación inicial**	**12**														
T1.1	Introducción al análisis de datos sociales	2														
T1.2	Introducción al Procesamiento del Lenguaje Natural	3														
T1.3	Introducción a la teoría de grafos	2														
T1.4	Introducción a topic modelling	3														
T1.5	Introducción al análisis de sentimientos	2														
R1	**Delimitación de scope**															

Table 2.1: First user story

2.2.2 Sprint #2

Identificador de la tarea	Nombre de la tarea	Puntos de historia	02/02/2023	03/02/2023	04/02/2023	05/02/2023	06/02/2023	07/02/2023	08/02/2023	09/02/2023	10/02/2023	11/02/2023	12/02/2023	13/02/2023	14/02/2023	15/02/2023
H2	**Data extraction**	**15**														
T2.1	Introducción a la API de Twitter	3														
T2.2	Extracción de datos de la API de Twitter	5														
R2	**Evaluación y próximos pasos**															
T2.3	Introducción al web scraping	2														
T2.4	Extracción de datos de Metacritic	3														
T2.5	Análisis y estructuración de los datos	2														

Table 2.2: Second user story

Initially, the second user story [2.2] was only going to be made up of tasks related to extracting data from Twitter, which would be carried out during a sprint starting on 27 March, so as not to coincide with other activities that would make the task difficult for the student. However, the transition of the API to a subscription-oriented payment model [34] made it necessary to carry out this phase in a somewhat more hurried manner, so in order to complete the data obtained in this way, it was also decided to collect data using web scraping techniques.

2.2.3 Sprint #3

The third sprint, in which the third user story [2.3] is carried out, is focused on the cleaning of the information obtained in the previous stage, seeking to provide the data with the appropriate form to carry out both a preliminary analysis of the data and a more thorough scrutiny based on the issues that are dealt with in them.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	10/04/2023	11/04/2023	12/04/2023	13/04/2023	14/04/2023	15/04/2023	16/04/2023	17/04/2023	18/04/2023	19/04/2023	20/04/2023	21/04/2023	22/04/2023	23/04/2023
H3	**Data cleaning y topic modelling**	**18**														
T3.1	Data cleaning	3														
T3.2	Análisis exploratorio de los datos	1														
T3.3	Topic modelling usando LDA	5														
R3	**Análisis de resultados**															
T3.4	Stemming y lemmatization	2														
T3.5	Pulido del modelo LDA	2														
T3.6	Topic modelling usando biterm	5														

Table 2.3: Third user story

2.2.4 Sprint #4

The fourth user story [2.4], developed throughout the fourth sprint, is focused on polishing the previous work, as well as putting into practice sentiment analysis techniques to obtain definitive conclusions about the products evaluated and the companies that offer them. The delay in the start of this sprint with respect to the previous one is due to the fact that the previous sprint had to be delayed as it coincided with the completion of mid-term exams by the student, which will be discussed in more depth in the time balance at the end of the chapter.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	04/05/2023	05/05/2023	06/05/2023	07/05/2023	08/05/2023	09/05/2023	10/05/2023	11/05/2023	12/05/2023	13/05/2023	14/05/2023	15/05/2023	16/05/2023	17/05/2023
H4	**Sentiment analysis**	**17**														
T4.1	Análisis de tweets mediante modelos preentrenados	3														
T4.2	Análisis de reseñas mediante modelos preentrenados	5														
T4.3	Análisis de reseñas mediante modelos propios	5														
T4.4	Reestructuración de los cuadernos de Jupyter	2														
T4.5	Conclusiones	2														
R4	**Presentación de la implementación final al tutor universitario**															
R5	**Presentación de la implementación final a la empresa**															

Table 2.4: Fourth user story

2.2.5 Sprint #5

Finally, in the fifth sprint, the fifth user story [2.5] is carried out, in which the final report of the work is written so that it can be presented to the final client. Although they are not included in the planning, constant communication is maintained with the tutor to provide feedback so that the written chapters of the document can be corrected.

Identificador de la tarea	Nombre de la tarea	Puntos de historia	17/05/2023	18/05/2023	19/05/2023	20/05/2023	21/05/2023	22/05/2023	23/05/2023	24/05/2023	25/05/2023	26/05/2023	27/05/2023	28/05/2023	29/05/2023	30/05/2023	31/05/2023	01/06/2023
H5	**Redacción de la memoria**	**23**																
T5.1	Introducción	2																
T5.2	Planificación	3																
T5.3	Contexto del trabajo	5																
T5.4	Data extraction	2																
T5.5	Data cleaning	1																
T5.6	Topic modeling	5																
T5.7	Sentiment analysis	3																
T5.8	Conclusiones	1																
T5.9	Apéndices	1																
R6	**Presentación de la memoria final al cliente**																	

Table 2.5: Fifth user story

2.3 Budgets

To calculate the cost of carrying out the project, we must distinguish two main sections: human resources (the cost of hiring and paying the workers in charge of carrying out the work) and technical resources (which include the set of hardware and software tools used for the final achievement of the Bachelor's Degree Project).

2.3.1 Approximation to cost estimation

Although, as mentioned above, it is not possible to make a direct conversion between the number of story points and the working hours corresponding to the achievement of the story, it is possible to make an approximation to this. However, this approach must always take into account the precepts of the agile methodology applied to cost estimation:

- Costs must be clear and transparent to all team members
- Costs need to be calculated automatically and frequently, so it would be useful to have a formula to facilitate this calculation
- Costs must be understandable and replicable

Based on these objectives, it is possible to define a formula that allows estimating the cost of each sprint, as shown in [58], on the basis of the following elements:

- $Horas \rightarrow$ Total hours devoted to the team's work
- $Salario \rightarrow$ Average hourly wage of workers in € per hour
- $Velocidad \rightarrow$ Average speed of the development team
- $PH \rightarrow$ Total history points of the sprint

We then obtain the following formula to calculate the estimated cost of each sprint:

$$Coste = \frac{Horas \cdot Salario}{Velocidad} \cdot PH$$

This formula is applicable to development teams made up of several people, but it is possible to adapt it to the framework of this one-person project if we consider the different roles that the student will have to adopt throughout the development. Specifically, there will be two such roles:

- **Data Scientist.** Responsible for the formulation of the questions and the extraction of the data that will lead to their solution. Therefore, you will be involved in both the development phase and the data collection phase. In Spain, their average salary is 35000 € per year [61], which becomes 2917 € per month prorated over 12 months (instead of dividing it into 14 annual payments) and represents a salary of 18.23 € per hour (considering a working week of 8 hours per day and 5 days per week; 40 hours per week and 160 hours per month).
- **Data Analyst.** They analyze the data obtained to obtain answers to the questions formulated by the data scientist, providing the information in a format suitable for drawing business conclusions. In our case, he will focus on the presentation of results as well as the writing of the final report. Their average salary in Spain is around 29000 € per year [60], which is 2417 € per month or 15.10 € per hour.

Hence, for the calculation of hours worked we will take the duration of each sprint (2 weeks working 5 hours a day; 70 hours) and only one member of the development team, whose salary will be the average of both roles taken on (16.67 € per hour) to simplify the calculations. In addition, as the average speed of the development team we will choose the average of the story points of all the defined ones (17 story points per sprint). This will allow us to estimate the cost of each sprint, as well as the total development of the project. For the latter, we must also consider the cost of registering the employee with the Social Security, which is 30% of their gross salary. We can see the final results in [2.6].

Sprint	Horas	Salario (€/hora)	Velocidad (PH/sprint)	PH	Coste estimado
Sprint 1	70	16,67 €	17	12	**823,69 €**
Sprint 2	70	16,67 €	17	15	**1.029,62 €**
Sprint 3	70	16,67 €	17	18	**1.235,54 €**
Sprint 4	70	16,67 €	17	17	**1.166,90 €**
Sprint 5	70	16,67 €	17	23	**1.578,75 €**
Coste total	**5.834,50 €**				
Coste total + Seguridad Social	**7.584,85 €**				

Table 2.6: Approximation to sprint cost estimation using story points

It should be noted that these values are merely a cost estimate prior to the development of the project. In order to see the actual final costs, it is sufficient to refer to the section 2.4, where the actual costs are calculated once the work has been completed.

2.3.2 Technical resources

The work has been developed on an Acer Aspire 3 laptop, with an AMD Ryzen 5 processor, 16GB of RAM and a 1TB SSD hard drive. This mid-range device is estimated to have about 4 years of useful life and cost 550 €, so we must calculate its amortization over the duration of the project (about 3 months, being 5 sprints of 2 weeks each). Apart from this device, a 1TB external hard disk has also been used to store a backup copy of both the downloaded data and the work itself. This component cost 50 € and is estimated to have an average lifespan of 8 years.

A stable Internet connection was also required to be able to consult sources of information, download the data to be analyzed or hold meetings with the project tutors, so it is necessary to include the cost of this in the budget. The chosen tariff provides 500MB of fibre at a price of 30.95 € per month.

With regard to software tools, both Jupyter Notebook and Overleaf offer free licences to work with them. In addition, Microsoft Office and Windows 10 operating system are included within the purchase of the laptop, so the reflected cost of everything will be counted as 0 € in [2.7].

Finally, although the project could have been developed before the change in Twitter's API business model, we have considered what the cost would have been if the development had been carried out after the change had taken place. Having downloaded almost a million tweets to be able to analyze them, it would have been necessary to use a month's Pro subscription to the API, which allows up to a million tweets to be obtained each month for a monthly subscription of 5000$ [71] .

Recurso	Coste	Porcentaje de uso	Total
Ordenador portátil	550,00 €	6,25%	**34,38 €**
Disco externo	50,00 €	3,13%	**1,56 €**
Conexión a Internet	30,95 €	300,00%	**92,85 €**
Twitter API Pro	4.661,95 €	100,00%	**4.661,95 €**
Jupyter Notebook	0,00 €		**0,00 €**
Overleaf	0,00 €		**0,00 €**
Microsoft Office	0,00 €		**0,00 €**
Windows 10	0,00 €		**0,00 €**
Coste total	**4.790,74 €**		

Table 2.7: Technical resources

2.3.3 Final cost

After analyzing both costs, it is estimated that the project would have a development cost of 7713.64€, which could rise to 12375.59€ if Twitter's paid API had been used.

2.4 Temporal and economic balance

The first *sprint* was completed ahead of schedule, mainly due to the fact that the task of introduction to graph theory was omitted, as the inclusion of these considerations would have lengthened the duration of the development too much and would have meant an overly ambitious scope of the project with respect to the teaching load of a Bachelor's Degree Project. The sprint ended with the meeting on 17 November, after 40 hours of work.

Although the tasks related to the extraction of data from Twitter were completed on time, mainly due to the time constraint underlying the cessation of free access to Twitter after the deadline, after the evaluation meeting, the completion of the web scraping tasks was postponed until mid-March. This was due to the fact that this stage coincided with the end of the student's internship and the start of the new university term. The work was carried out from 17 March to 24 March. The total number of hours dedicated to this work was 70.

Again, the tasks leading up to the third meeting were completed on schedule, although both the meeting and the subsequent work were postponed until 3 May, as it coincided with the university's mid-term examination period for the student. The sprint was concluded on 8 May, after 75 hours of work.

Immediately after the completion of the previous sprint, the fourth sprint began, seeking to balance the time mismatch suffered and thus be able to keep to the originally stipulated deadlines. This effort paid off and the sprint was completed on schedule, after 65 hours of work.

The biggest problem arose with respect to the last sprint, as it was not possible to keep to the objective set, as the final university exams began and the respective final practicals of these subjects had to be handed in. This casuistry, together with other reasons of a personal nature, meant that the writing of the report had to be done at irregular intervals, culminating on 30 June, with a total of 90 hours of work.

[2.8] shows the final Gantt chart, with the final development dates for each of the tasks that make up the various user stories mentioned above.

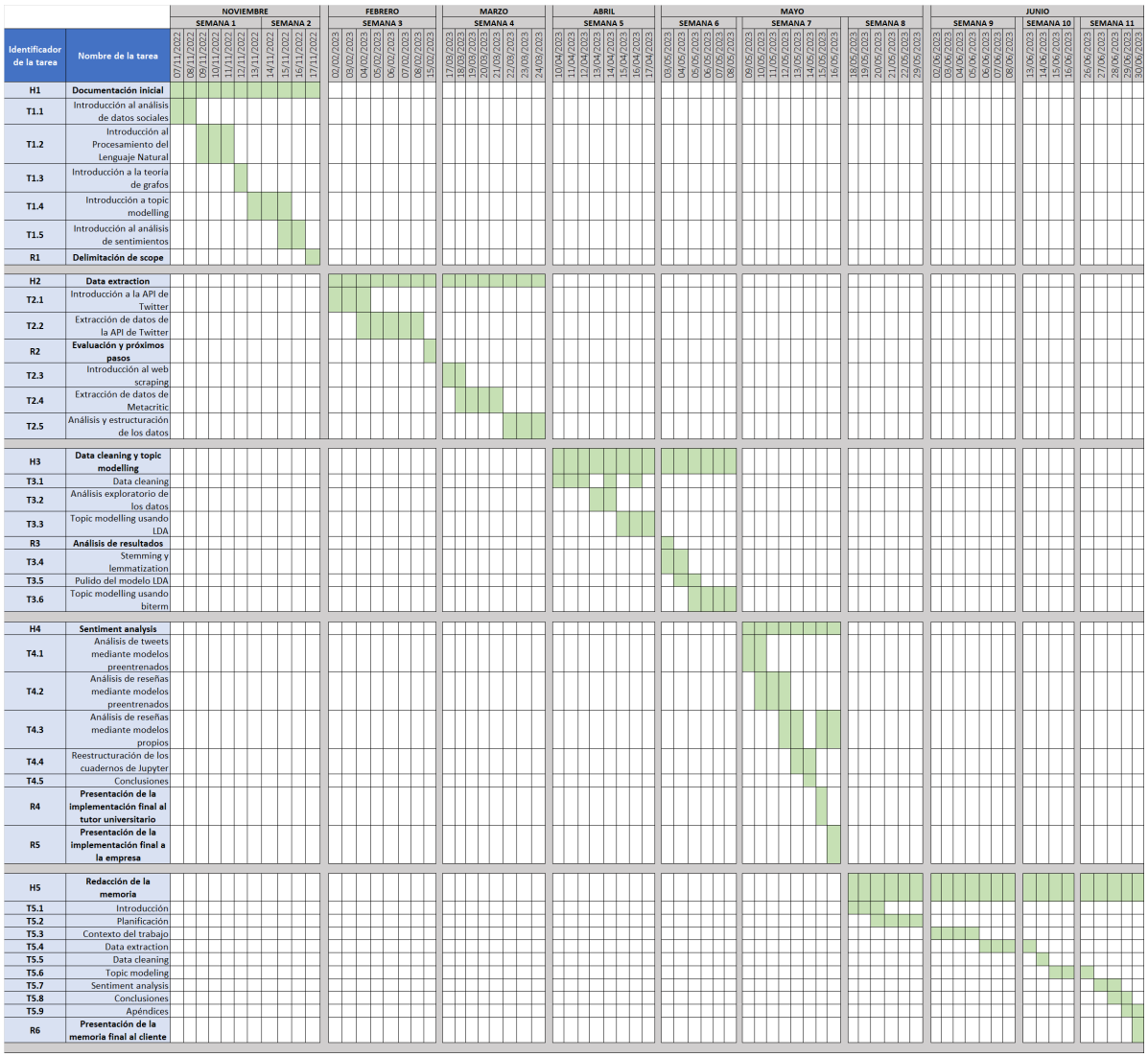

Table 2.8: Final Gantt chart with the dates on which the project was carried out

After the time balance, we can **adjust the economic budgeting carried out using the total hours worked by the student as a reference**, instead of using the estimate previously given. The total number of working hours was 340, which is in line with the teaching load of the Bachelor's Degree Project (between 300 and 360 hours as it is 12 ECTS). Considering that the salary of the worker is the aforementioned 16.67 € per hour, this means a total of 5667.80 € to which must be added the cost of Social Security: a total of 7368.14 €, very similar to the estimate given, which shows the robustness and precision of the chosen estimation method. In conclusion, the final budget for the work has been 7496,93 €, which could have reached 12158,88 € in the case of having needed to use Twitter's payment API.

Chapter 3

Context of the work

3.1 Specific environment

The **environment** of a company is the set of factors external to an organisation that, although having a significant influence on its strategy, the organisation cannot control. Within this, we distinguish the **general environment**, which is common to any organization and is determined both by society and by the nature and characteristics of the socioeconomic system; from the **specific environment**, which is specific to the task or activity characteristic of the organization [53].

As mentioned in the introduction, this work is the result of a collaboration with the company NielsenIQ, so it will be interesting to analyze the specific environment of the company in order to be aware of the weaknesses and strengths of the commissioned study in relation to the rest of the market.

3.1.1 Company-specific environment

NielsenIQ is a market research and data analysis company, offering companies the opportunity to analyze consumer behaviour in detail and to adapt their marketing strategies based on these trends.

In order to understand where NielsenIQ stands in the data analytics and market research industry, we will apply **Porter's competitive forces model** [3.1] to better understand its position:

1. **Competence:** NielsenIQ operates in a highly competitive and evolving market with a number of companies offering similar services. Direct and potential competitors that NielsenIQ faces in terms of technology, research methodologies and geographic coverage include [48]:

- **Syndigo.** NielsenIQ offers its clients an integrated platform for the collection, management and analysis of their own data. However, NielsenIQ has a competitive advantage over its clients thanks to a better customer service, as well as recognition from authoritative industry bodies such as the FDA (Food and Drug Administration) or the AHA (American Hospital Association).
- **The Data Council.** A company that provides retailers, primarily those selling consumer packaged goods, with clear, accurate, independent and complete information about products in their supply chains. However, NielsenIQ again shows its superiority in the market by offering its clients a higher quality service, with more consistent datasets, as well as greater scalability options depending on the client's needs.
- **Spins.** A provider of retail consumer information, analysis and consultancy for the natural, organic and speciality products industries. As with the other competitors already mentioned, the user experience offered by NielsenIQ and the improved scalability alternatives put NielsenIQ at an advantage.

Although NielsenIQ holds a privileged position in the data analysis sector, particularly for retail, the emergence of new technologies and the constant evolution of analytical tools forces the company to constantly improve in order to maintain its quality standards and not lose its privileged position in the sector.

2. **Suppliers:** The raw material that NielsenIQ works with is data, so it is necessary to have a wide range of suppliers to guarantee the quality and availability of the data. Some of these suppliers are:

   - **Retail data providers.** Information on product sales in different categories and geographic locations, obtained from strategic agreements and partnerships.
   - **Audience data providers.** Audience and media consumption information across different channels such as television, radio and digital media. From this viewing and interaction data, NielsenIQ can perform the relevant measurement and analysis.
   - **Demographic data providers.** Information on the demographic, socioeconomic and geographic characteristics of the population, enabling the company to better understand consumer behaviour.

   In addition, the company also needs **technology and software providers** to be able to develop and maintain its data analysis and visualization platforms, as well as to obtain tools that enable it to collect, analyze and present data efficiently and effectively.

3. **Customers:** NielsenIQ works with a wide variety of clients, most notably consumer goods companies and retailers, to enable them to analyze the performance of their products and understand consumer behaviour, allowing them to tailor their strategic decisions to the realities of the current context. In addition, NielsenIQ also serves media and advertising companies, enabling them to evaluate the reach and effectiveness of their campaigns, as well as other research and consulting agencies that provide services to their own clients, giving them the opportunity to support their own research, analysis and business strategies.

4. **Substitutes:** In addition to the aforementioned alternatives within the industry itself, clients seeking NielsenIQ's services can opt for in-house data solutions through the creation of a dedicated in-house analytics department. However, this would require a high investment in both infrastructure and personnel, although even such an investment would not guarantee solutions of the quality offered by companies focused solely on this kind of research.

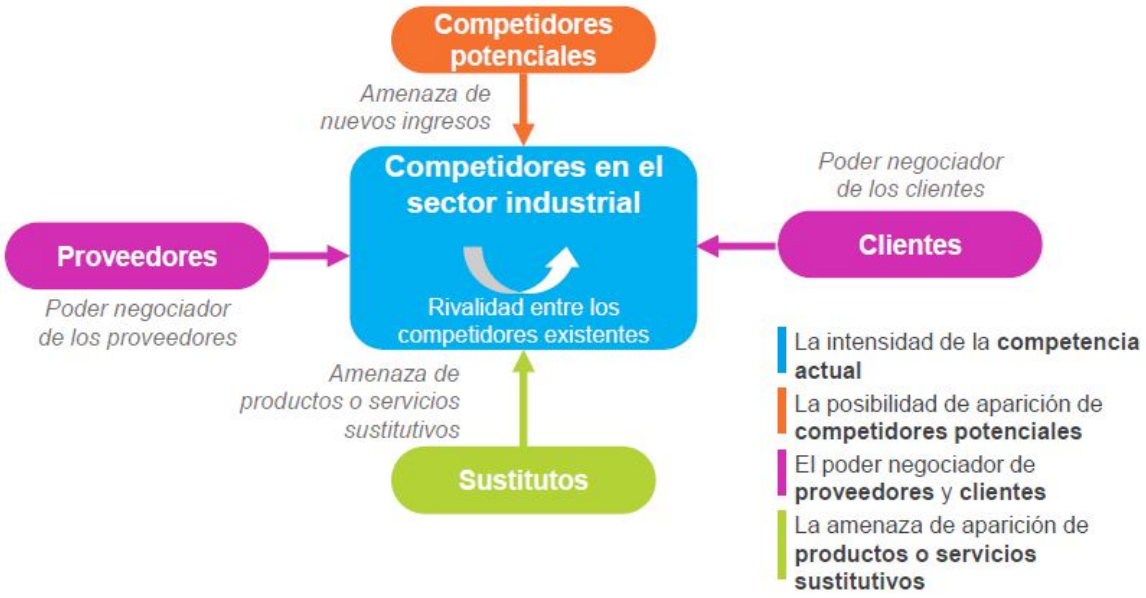

Figure 3.1: Porter's model

The data analytics industry can already be framed as an **established strategic sector**, as it is a key factor in planning the future strategy of any company. It is also a **fragmented sector**, as the market research industry is highly competitive; as well as an **emerging** one, due to the rapid technological advances that the sector is experiencing (such as big data analysis or artificial intelligence).

3.1.2 Specific environment of the study

As explained in the data science methodology, the first step is to clearly and precisely define the subject of the study to be carried out. For this work, we will study the presence and impact on social networks of the three main video game companies at present, which offer both hardware and software: Nintendo, PlayStation and Xbox. Specifically, the topic chosen facilitates the comparison to be made because, when the data was collected from the social network Twitter (first week of February), the three companies had just released three new games developed by the companies' own studios: Fire Emblem Engage, Forspoken and Hi-Fi Rush, respectively. It will then be possible to compare the reception of the games by the general public and the trade press, gathering information from review pages and social media impressions.

Before carrying out the study, it seems appropriate to give an overview of the current situation of the video game industry by analyzing the position of the three companies in the market. While in the early days the industry was quite fragmented, with several companies vying for the public's favour and choosing their console over the competition, today big names such as Sega and Atari have been relegated to a more secondary role and focus solely on software development. Leaving aside the undeniable rise of the PC as a gaming platform or consoles derived from that model such as Steam Deck itself, when a consumer considers purchasing a new video game console, he or she has three main alternatives in the current generation:

- **Nintendo Switch:** It is the Japanese company's seventh desktop console. After the disappointment of the Wii U, its previous console, the Switch is a hybrid model that allows gamers to combine classic desktop gaming with the features of a handheld console. Despite the performance limitations that come with it, the console has had a broad appeal, establishing itself as a crowd favourite and is particularly suited to gaming with family and friends. Since its launch in March 2017, it has already become the third best-selling console in history, surpassing the company's own successes such as Game Boy and Wii [41].

- **PlayStation 5:** As the name suggests, it is the fifth model in the PlayStation family, a brand owned by Sony Interactive Entertainment. The Japanese company launched the console in November 2020, but the global pandemic and the shortage of components for the manufacture of consoles has limited the available stock, forcing a large part of the releases to be intergenerational or to also be released on PC. However, this generational transition prompted a change in the company's model, adapting its monthly subscription plans to be specifically designed for PlayStation 5 users. In particular, the games offered through the PlayStation Plus service rotate at the beginning of each month.

- **Xbox Series X:** Like Sony's console, Series X was launched in November 2020, although it did not suffer from as many stock issues as its direct rival. Microsoft, however, is more in favour of a combined business model, releasing all its games simultaneously on both console and PC. This is mainly due to its Game Pass subscription service, which in 2022 surpassed 25 million users [24]. Xbox Games Pass offers its subscribers an extensive catalogue of games, which is expanded each month with new releases, giving them the chance to play titles on both console and PC.

The most direct competition lies between PlayStation and Xbox, as both companies have made the generational transition at the same time and are trying to take control of the market. Nintendo, on the other hand, with its hybrid model and more familiar experiences, already has its own niche market and can establish some coexistence with the other two companies. At the end of January, all three companies released games developed by internal studios and focused on individual user experiences, so the comparison between them seems appropriate. As a final note, we will also give some context on the environment and circumstances in which the games have been developed:

- **Fire Emblem Engage:** The seventeenth instalment in the acclaimed tactical role-playing franchise Fire Emblem, developed by Intelligent Systems. Following the success of the previous game in the series, Three Houses, and the franchise's international expansion outside of Japan with products such as Fire Emblem Heroes, it has a large and established fan base around the world. In the game, we play as Alear, scion of the divine dragon, who must face the threat of Sombron, the shadow dragon who plans to overrun the continent.

- **Forspoken:** Developed by the studio Luminous Productions, a subsidiary of the video game company Square Enix, this game is their first solo project following their collaboration on the development of Final Fantasy XV. The game tells the story of Frey, a young New Yorker who ends up in the beautiful and cruel world of Athia. As she figures out how to return home, she must use her newfound magical abilities to traverse vast landscapes and confront monstrous beings.

- **Hi-Fi Rush:** Shinji Mikami, creator of the Resident Evil video game franchise, founded his own video game studio in 2010 away from the Capcom umbrella. In a radical departure from his previous work, Hi-Fi Rush tells the lighthearted story of Chai, a boy who dreams of becoming a rock star and who, after a freak accident, sees his heart replaced by a music player that enhances the powers of his robotic arm when he moves to the beat of the music. This curious rhythmic hack and slash platformer is the first game from the studio, Tango Gameworks, and is published by Bethesda Softworks. The game was immediately announced and released in a Microsoft presentation, adding that Game Pass subscribers would also be able to play it at no extra cost, due to the US multinational acquiring Bethesda in 2020 as part of its strategy to encourage the development of exclusive games [67].

3.2 Business environment

This section provides an introduction to social data analysis, giving a basic terminology of the theory underlying the field, as well as some current applications of such studies.

3.2.1 Social data analysis

As mentioned in the introduction, today we generate more and more information on a daily basis, be it on social networks, the Internet in general or even through devices belonging to the so-called Internet of Things (IoT). Given the vast amount of data at our disposal, it is worth asking ourselves what conclusions we can draw from it, as much of this data may have no relevance at all. It is therefore necessary to first identify what terms such as "knowledge", "usefulness" or "relevance" refer to.

Traditionally, definitions of knowledge come from the field of information science. The concept of knowledge is usually represented as a pyramid known as the **knowledge hierarchy**. This pyramid is built on data, allowing information to be built from it, leading to knowledge itself.

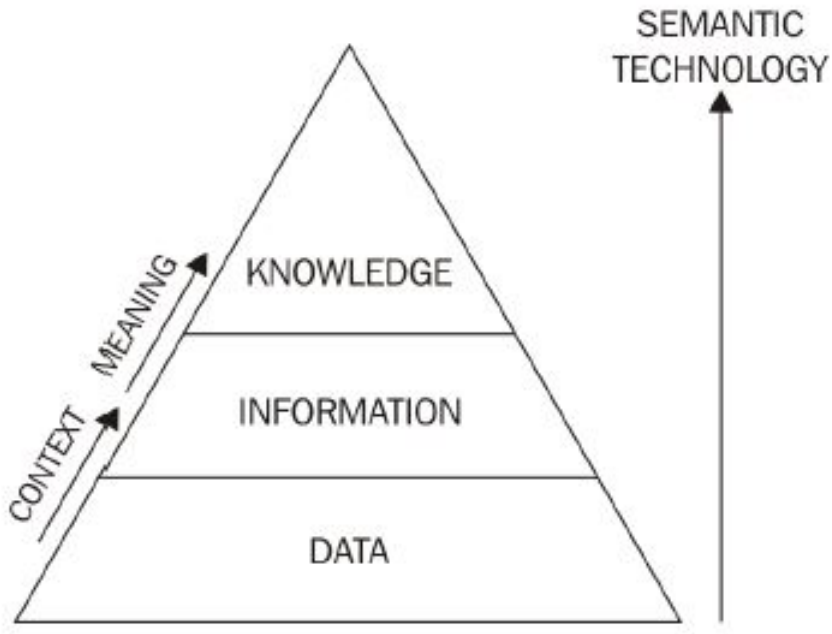

Figure 3.2: From raw data to semantic information

Therefore, scaling this pyramid is about moving from raw data to knowledge, which is achieved by analysing the context and meaning of the data. In this way, the technology we build gains a deeper understanding of the original data and, more importantly, of the users who generated it. In other words, it becomes useful.

In this context, **useful knowledge** is knowledge that can be understood as **actionable** knowledge, which is knowledge that enables decision-makers to implement business strategies in a justified manner [15].

The dataset extracted from social media is known as **social data**, which can be either **structured** if it is numerical or quantifiable, or **unstructured** such as video or images, which are a great source of information but are more difficult to analyze. The process of applying rigorous methods to make sense of social data is called **social data analytics** [17], ranging from the construction of models to understand and analyze data, to the visualization of the results obtained through these models.

In the section 2.1.1 we have already gone into the different steps that must be followed for the correct planning and execution of data science projects, the discipline under which social data analysis is framed, so for a detailed analysis of the process, it is sufficient to refer to that section.

3.2.2 Cases of application

Let us now look at some examples of the application of data analysis techniques. As these methodologies can be extrapolated both to general analysis and to that applied specifically to social data, we will present examples from both cases to see the power and versatility of these techniques:

- **Politics.** 2012. Barack Obama is running in the US general election in an attempt to revalidate his mandate after having been President for the last four years. The main difference with respect to his previous campaign is that now his data analysis team is made up of 20 people, thus quintupling the number of staff in this area with respect to the previous elections. Thanks to this, Obama's campaign team is able to compile all the information that US citizens post on the web, identifying which issues are most important to his voters and thus influencing them during the campaign; or capturing the votes of the undecided thanks to advertising messages in spaces that they can easily see, such as the commercial breaks of the series "The Walking Dead" or in threads on the Reddit forum [46].

  Using these methods, Obama won in key states such as Ohio, and managed to retain his position as President of the United States of America. And all of this, supported by the use of social data analysis and data-driven strategies.

- **Sports.** One of the most famous recent stories in English football is when former Liverpool analyst Ian Graham met for the first time the manager who had just signed for Liverpool that season: Jürgen Klopp. In that meeting, Graham spoke to Klopp about some of the German manager's games last season with his former team, Borussia Dortmund, in which the Bavarian side should have beaten their opponents comfortably, but factors such as luck and strong performances from their opponents led to adverse results for Klopp's side. Seeing Graham's extensive knowledge, Klopp was surprised at how many matches the English analyst must have watched.

However, Graham had not seen a single one of the games: he had only relied on the reports generated by the analysis team to get a full understanding of what had happened in each match [56]. As a result, the German manager was convinced of the power of this innovation and was open to the data analytics team's advice on the development of his squad as well as on the improvement of his playing systems. This collaboration proved to be extremely fruitful, and today the Merseyside team has regained its status as one of Europe's top teams, winning a league title and a European Cup along the way.

This story is a perfect example of how even institutions that were once as stagnant as football teams, stuck in the classic methods and systems that had worked over the years, have benefited from innovative approaches such as data analysis.

However, this methodology has proven to be an effective model not only for the big European teams, but also for the smaller ones. By applying these tools, teams such as Brentford and Brighton, whose chairmen own data analytics companies and use this methodology in their own clubs, have managed to climb from the lower divisions of English football to make it into the top 10 of the Premier League this season. All this, thanks to the use of data analysis to sign their players or to find weaknesses and possible improvements in their playing systems [12].

- **Retail analysis.** As seen in the 3.1 section, retail can derive great benefits from data analysis, allowing companies to adapt to market trends and meet new consumer needs. This methodology can also be extrapolated to big brands, such as Amazon, which is able to offer personalized product recommendations that are more in line with consumer preferences based on consumer data.

- **Medicine.** Big data analysis can also be useful for monitoring and predicting the evolution of epidemics and disease outbreaks. Through the use of graph theory, it is sufficient to identify the most influential nodes in a network to limit the spread of disease as much as possible. Such use can be seen in studies such as [5] which, among many other applications, discusses contact detection and tracking to minimize the spread of COVID-19.

3.3 Scientific-technical context

3.3.1 Machine Learning

Within Artificial Intelligence, Machine Learning is the discipline responsible for the study and development of algorithms that, based on data, can make predictions about the data itself. In other words, it is based on making predictions based on known properties of the data [15].

Programmes based on the use of Machine Learning are a constant in our daily lives, ranging from simple tasks such as deciding whether an email is spam or not, to more delicate ones such as analyzing bank transactions in order to identify possible fraud attempts.

Broadly speaking, we can distinguish two main categories within the most popular Machine Learning methodologies. Although this is a simplification that is not able to encompass the full depth and breadth of Machine Learning, it is a good starting point to appreciate some of the technical aspects of Machine Learning.

- **Supervised learning.** This methodology is used to solve classification problems, where the data contains additional attributes other than the one to be predicted. In the ideal scenario, the model built would associate the expected output with each input, without any possibility of failure. To do this, a mathematical model is built using test inputs that will serve as a training set, so that the model then attempts to infer the desired attribute, called a class or label, from a different set of data from those used for training (test dataset). The most common supervised learning techniques are Naive-Bayes, Support Vector Machines (SVM) or models belonging to the neural network family, such as perceptrons or convolutional networks.

- **Unsupervised learning.** In contrast to the previous case, unsupervised methods are used in problems that have an unlabelled dataset as input, so the final result is unknown. The typical example of this kind of challenges are the so-called clustering problems, in which we try to find underlying patterns under the data and thus be able to divide them into groups, or detect those elements that do not share similar characteristics with the rest (outliers detection). Examples of this kind of algorithms would be k-means, k-medians or Kohonen maps.

3.3.2 Natural Language Processing

Natural Language Processing (NLP) is the discipline concerned with the study of methods and techniques for the analysis, understanding and generation of natural language; that is, language spoken or written by human beings [15].

This definition inevitably evokes the origins of Artificial Intelligence (AI), when Alan Turing presented his test of the same name in 1950 to discern whether or not a machine possessed intelligence. In it, a person had a written conversation with two agents, a human and a machine. If the evaluator was unable to distinguish whether the answers given to a series of questions had been given by the human or the machine, it was determined that the machine possessed intelligence. The main complexity of the test is that the computer must possess natural language processing, knowledge representation, reasoning and machine learning skills to be able to pass the test. We can therefore situate NLP as a branch within AI that, based on the use of automatic learning techniques and knowledge of linguistics, allows us to analyze, understand and even emulate the ways in which human beings express themselves.

Despite the emergence of conversational bots and highly advanced artificial intelligences such as ChatGPT, when we try to have a conversation with them, we can still see certain rough edges that show that our interlocutor is not really a human being. This is why natural language processing remains a booming area of research today, with more and more possible applications in various fields.

In the context of social media, it is evident that there is a large amount of textual information waiting to be collected and analyzed. The amount of information is constantly increasing every day, whether in the form of social media conversations or product reviews given by users. However, the transition from raw data is a difficult task. Fortunately, through the use of NLP techniques, we may be able to discover the main topics that are being talked about on social media or identify the sentiment associated with reviews on a review page.

Natural language processing is composed of a wide variety of tools and methodologies. That is why, throughout the work and as we use them, we will present the different theoretical bases in each of the steps necessary for the analysis of social data from an NLP perspective.

3.4 State of the art

As a theoretical study on a specific topic, it is difficult to make a direct comparison with related work. Instead, there is a wide variety of articles and publications where different phases of the social data analysis process are carried out. Therefore, a number of studies whose content corresponds directly to different parts of this paper are presented below, in order to observe the methodology and tools used by the authors, as well as the possible conclusions that can be drawn from them.

3.4.1 Web scraping of Metacritic

Project by data scientist Martín Pellarolo, published in 2018 on his GitHub page [52]. The work is based on the analysis of video game data obtained from the review site Metacritic. More specifically, it obtains detailed information and reviews of games available for Nintendo Switch, PlayStation 4 and Xbox One, the predominant consoles of that generation. The study consists of four Jupyter notebooks:

1. **Data scraping.** Using the DOM format that the Metacritic pages follow, the set of games available for each platform can be extracted. To do this, a list of all the titles is extracted from the main page and then individual information is obtained for each of them (press and user reviews).

2. **Data cleaning.** The aim of this notebook is to sift out non-relevant information and to unify all data in a single format, so that the information collected can be easily worked with.

3. **Exploratory data analysis.** Comparative study between the different consoles and their releases, analyzing data in terms of exclusives, top-rated games or the distribution of games on each platform in accordance with the age rating system (see figure 3.5). General data is also analyzed, such as the time of year when the most games are released or whether there is a big difference between critics and

users ratings (see figures 3.3 and 3.4). An exploratory analysis of the data is carried out to try to uncover its main characteristics.

4. **Sentiment analysis.** Creation of a model for the classification of reviews based on their content, to determine whether they are positive or negative. To this end, two different models are studied for its implementation: multinomial Naive-Bayes and convolutional networks.

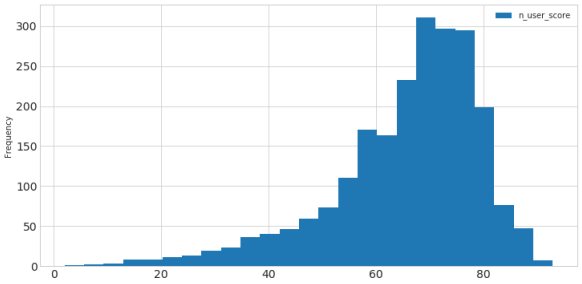

Figure 3.3: Press ratings on Metacritic

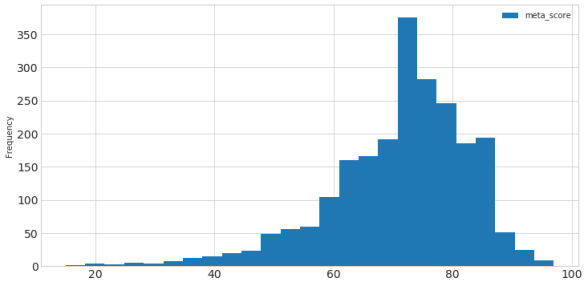

Figure 3.4: User ratings on Metacritic

The similarities between Pellarolo's project and the one proposed in this paper are evident. However, our study focuses more on the analysis of data from social networks, considering the data through Metacritic as one more source of information to work on. For this reason, we will not delve as deeply into the analysis of reviews as Pellarolo does in his work.

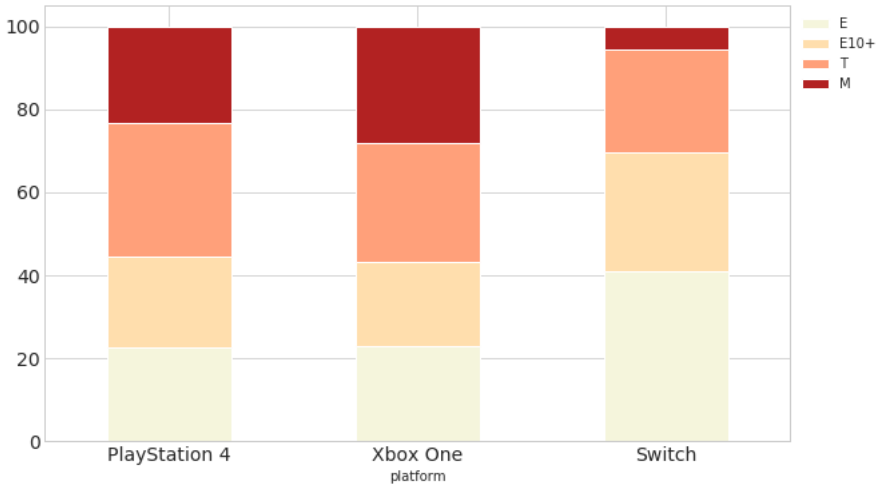

Figure 3.5: Proportion of games by age rating for each console

3.4.2 Topic modelling of a set of tweets

The website Medium is an American online platform that is one of the leading exponents of **social journalism**, pages that combine the publication of articles by users with others written by specialised professionals, the latter being a way of encouraging visitors to subscribe so that they can access similar content. Within this second group is an article by French computer engineer Clément Delteil, who specialises in AI. In his article, originally published on the specialised website Towards AI, Clément presents a step-by-step guide to analyse a set of half a million tweets about Elon Musk [19].

Throughout the article, it is shown how to build a Google Colab notebook to run the whole study in the browser itself, implemented using Python. After performing a simple pre-processing of the data, the most frequent terms are analyzed in order to extract the predominant themes in the text and, once obtained, also perform a basic sentiment analysis on them. The main strength of this study with respect to ours lies in the analysis of communities and entities since, as can be seen in figure [3.6], it is possible to obtain accounts mentioned in the tweets and even analyze the sentiment associated with them. However, such analysis is possible when dealing with real entities or organisations, but as our study analyses specific products -for which it would be unfeasible to create a specific official account for each one- such analysis is not possible here. Moreover, the brevity of the article forces the author to limit the length of the topics addressed, which results in a more superficial analysis.

Figure 3.6: Wordcloud with the most mentioned user accounts in tweets

3.4.3 Sentiment analysis of Amazon reviews

Once again, on Medium, we find the study conducted by data scientist Enes Gokce, from Penn State University. In the work originally published on the specialized website Towards Data Science, Enes conducts a sentiment analysis on a set of Amazon product reviews [28]. The analysis is developed over the course of a Jupyter notebook, which together with four others forms a complete study project using NLP, which is available on his GitHub page [27].

However, both the notebooks and their final presentation are mainly focused on sentiment analysis, so it is this case that we will analyse here. Based on the data obtained, which have already been sifted and analyzed in the two previous notebooks, an analysis is made of the reviews obtained and compared with the scores given. Factors that at first sight might not seem so relevant are also analyzed, such as the length of the reviews themselves in the final perceived evaluation of the text (see figure 3.7). In addition, the influence of subjectivity is also considered and the extreme cases present in the analysis are analyzed.

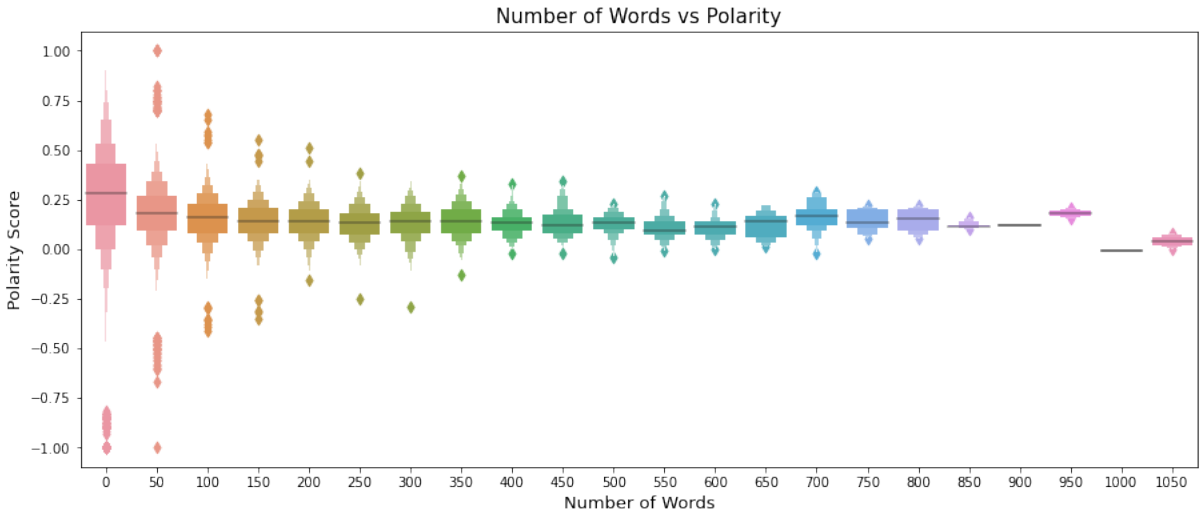

Figure 3.7: Rating of reviews according to their length

As Gokce himself comments, the data cleaning was somewhat superficial, so the use of techniques such as stemming or lemmatization would allow more accurate results to be obtained. Finally, the notebooks do not show how to obtain the data used for the analysis, which is a fundamental part of any data analysis project.

Part II

Development of the proposal and results

Chapter 4

Data extraction

4.1 APIs

An **Application Programming Interface** (**API**) is the way in which data is exchanged between a service and the developer or user. In the context of social media, it allows to share data with third-party application developers. However, the popularization of data science has turned APIs into a way to effectively mine information for knowledge creation [17].

Although each social media is implemented differently with its own particularities, there are currently two main types of APIs:

- **RESTful:** The most common type of API provided by social media. The information it collects is static and is retrieved from historical data. **REST** (Representational State Transfer) is a software architecture that imposes a number of restrictions on communications based on the HTTP protocol for data transfer. The most important methods are GET and POST, which allow the retrieval and publication of data from distant machines, respectively.

- **Stream:** It allows data to be obtained in real time. The result obtained is practically the same as the historical data.

The use of an API makes it possible to obtain social data for commercial and marketing purposes, or gives developers the possibility to integrate the social media in question into their projects. However, the use of APIs has a number of **limitations** that need to be taken into consideration:

- **Rate limits.** Due to infrastructure and business constraints, companies limit the amount of data entering or leaving their systems. Therefore, this forces users to do careful planning when obtaining data.

- **API changes.** As they belong to private entities, these can modify or limit access to their API at any time, which forces data analysis experts to be prepared to adapt to possible future cases. The clearest example of this is the change to a payment model that Twitter's API underwent after Elon Musk bought the company [34].

- **Legality.** The use of API data that does not comply with the company's usage policies can lead to legal action, so it is vital to adhere to these frameworks when developing projects using these tools.

Connecting to an API

In order to be able to connect to the API of a social media, it is necessary to make a prior configuration, which depends on the platform in question. However, the general process can usually be summarised in the following steps:

1. **Registration in the application.** Provide personal information and information regarding the purposes for which you plan to use the API. Once this is done, keys are generated to be able to access the API functionalities, called **authentication keys** or **consumer keys**.

2. **Authentication** through the authentication keys generated in the previous step.

3. **Search for endpoints.** Depending on the provider, the endpoints of each API vary, so it is necessary to carefully read the documentation provided by the companies to identify the endpoints that will be needed during the development of the project.

Of all these steps, the one that can be the most complex is **authentication**. Fortunately, today this process has been unified for almost all platforms thanks to the use of OAuth.

OAuth is an authorization protocol that allows users to share data with an application without the need to use their password. Its main advantage is that it allows access to third parties based on the limitations of their tariff, establishing a standardized and secure connection. It can be done as a user or as an application (without the need for user context). For data extraction, the latter case is the most frequently used, and requires several previous steps before any kind of information can be obtained:

1. **Creating a user or developer account.**

2. **Creating an application.** Once you have an account, you can access the control panel or developer console, which gives access to all the functionalities for managing the account in question, as well as creating and deleting other applications or checking the use of quotas by the application. To access this dashboard, it is necessary to have first created an application. Figure 4.1 shows the dashboard of a Twitter developer account.

3. **Obtain access tokens.** Generate the access tokens of the application created in order to be able to connect to the API afterwards. Depending on the company, it may be necessary to specify additional permissions within the HTTP requests made or to indicate what will be the scope of the actions performed before generating the access tokens.

4. **API connection.** Once the tokens have been obtained in the previous step, requests can be made, always taking into account the quota limitations for each platform.

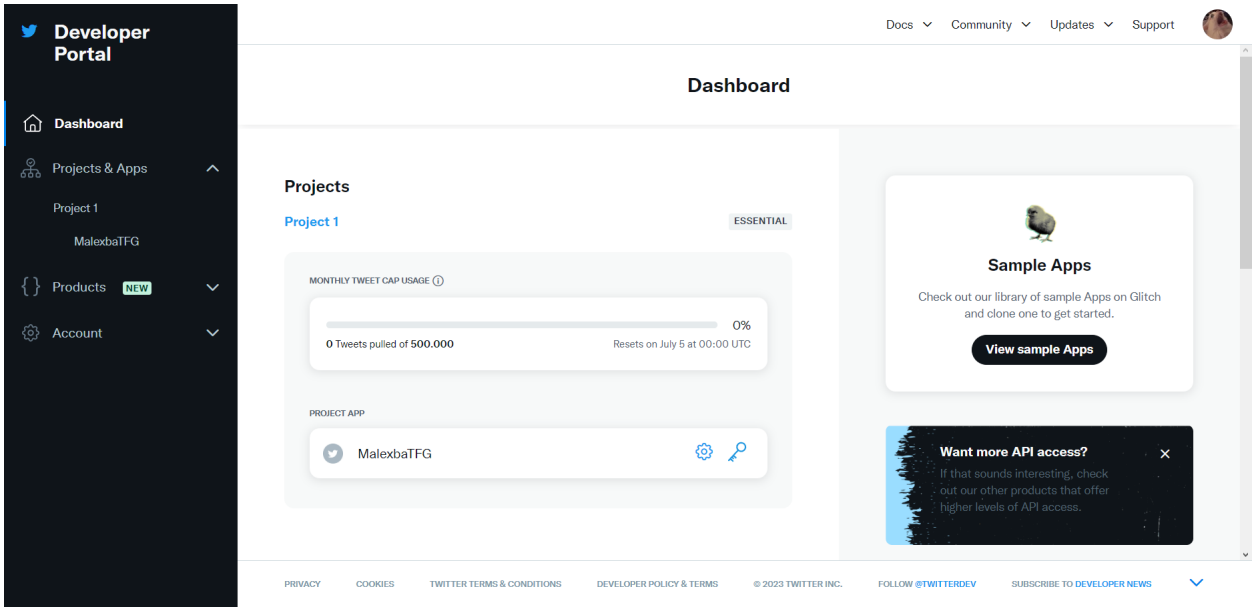

Figure 4.1: Dashboard for a Twitter developer account

4.1.1 Twitter

Microblogging is an online service through which users can share messages, links to external sites, images or even videos, so that they are visible to all subscribers to the service. In contrast to traditional blogs, microblogging is mainly based on short texts. Some of the best known are Tumblr and Mastodon, although the most popular platform for microblogging today is still Twitter.

Launched in 2006, Twitter has more than 200 million active users every day, with an average age of around 30. It is particularly popular in the United States and Japan, with 120 million users in these two countries alone. According to statistics, the average Twitter user spends an average of 6 minutes on the application, usually to search for news or to keep themselves entertained [18]. Due to the ease of accessing and publishing posts, it

is one of the largest datasets of user-generated content. Within this context, we need to identify the basic terminology within Twitter [26]:

Tweet Every message posted on Twitter. Its content can be images, videos or links to other pages, although it tends to be mainly text. Originally there was a 140-character limit, which has now been extended to 280 characters. However, Twitter Blue users can post messages up to 10,000 characters long.

User A user must be registered in order to be able to post messages. At the time of registration, it is necessary to indicate a user name that will be associated with the messages posted.

Mention You can mention another user in a tweet by specifying the username of the user you want to mention after an @ (*@user_mentioned*).

Reply A tweet marked as a reply is linked to another tweet, allowing the creation of **threads** (chains of tweets linked as replies). It is also indicated by mentions of who is being replied to. Depending on the privacy options selected by the author of the original tweet, other users may or may not reply to their tweets.

Follower User following another user, as well as their activity. Notifications can also be activated, to receive immediate information about another user's activity.

Retweet Redistribute a tweet. Allows the user to republish the tweet on their profile, maintaining the original attributes of the tweet. Within this command, there is also the option to mention the tweet, with the user creating a new one that maintains a reference to the original tweet being discussed.

Like A method of interaction to indicate that a post is liked by the user, promoting its visibility to other Twitter members.

Views Counter with statistics on the number of people who have seen the tweet, as well as more specific statistics on whether the post has led to a profile visit or how many times a multimedia content has been played, if attached.

Hashtag A method of tagging posts according to topics. They are generated spontaneously by users and make it possible to increase the visibility of posts by grouping them by topic and making them easier to find.

Timeline Each user's homepage where they can see different tweets. Since 2023, it has been divided into the "For you" section, which shows tweets that the Twitter algorithm considers to be of interest to the user, and "Following", which shows the activity of followed users.

User profile Main page of the user, where you can see data such as their name, a brief description of their profile or the accounts they follow/followed. It consists of four subwindows: "Tweets" (includes retweets), "Tweets and replies" (shows only the posts made by the user), "Multimedia" (which includes all the posts that include multimedia content) and "Likes" (includes all the tweets that the user has liked).

Privacy Twitter options on visibility to the world. Apart from whether a profile is private or not, the concept of circles was also included, which allows messages to be published so that only selected users can see them.

4.1.2 Twitter API

Twitter offers developers several different APIs: the Rest API, the Streaming API (which allows you to get tweets in real time) and the Ads API. This document will focus on the use of the first one. Detailed documentation, as well as usage guides, can be found at [20]. When making requests, it is important to take into account the specific rate limits for each command, which can be consulted at [70].

In order to use the API, it is necessary to follow the connection and authentication steps indicated in 4.1. In this case, to connect to the API, we will use the Python library **Tweepy** [69], which allows easy access to the Twitter API.

Tweepy uses **OAuth2** as its authentication model, which shares the goals of its previous version, but has been built completely from scratch. To connect, simply instantiate a Tweepy client by specifying the application's bearer token and the format in which you want to obtain the data (in our case, as a dictionary). When making requests with the corresponding method, you must specify the fields you want to obtain in each request in terms of tweet, user, location and multimedia content. However, since many of these are optional or may not be accessible for privacy reasons, the only fields that will always be obtained are those of the tweet. In addition, if tweets are to be fetched, paging options must be specified, as the downloading of tweets must be done in successive requests of a limited size to avoid exceeding the rate limits and being considered abusive users. As an example, a request made with the options specified in 4.1 would return a total of 50,000 tweets, if enough posts had been generated in the one-week timeframe to which the basic Twitter API plan gives access.

```
import tweepy
# Creation of the client
bearer_token = "aqui_iria_el_bearer_token_de_la_aplicacion"
client = tweepy.Client(bearer_token = bearer_token, return_type=
    dict)
# Define the fields to be obtained
tweetFields = ['id','text','edit_history_tweet_ids','attachments','
    author_id','context_annotations','conversation_id','created_at',
    'entities','in_reply_to_user_id','lang','possibly_sensitive','
    public_metrics','referenced_tweets','reply_settings','source','
```

```
    withheld']
userFields = ['id','name','username','created_at','description','
    public_metrics','verified']
placeFields = ['full_name','id','country','country_code','geo','
    name','place_type']
mediaFields = ['public_metrics','alt_text','type','url','variants']
# Page setup
nextToken = None # Token para encadenar las peticiones
maxResults = 100 # Maximo de resultados obtenidos por peticion
pagTimes = 500 # Numero de peticiones que se realizaran
```

Code 4.1: Tweepy client configuration example

4.1.3 Selected case study

Although the general idea for the Bachelor's Degree Project was provided by the company, the choice of a specific topic of study was left up to the students themselves. Initially, the possibility of choosing as the object of study companies that are highly involved in having a presence and relevance in social networks through their Community Managers, as is the case of KFC, was considered. However, when starting the process of extracting information through the Twitter API, it was observed that a maximum of tweets up to one week old could be collected. This time limitation led to think about current topics on which to focus the collection of information, a process that was carried out in the first week of February 2023, giving rise to two fundamental themes:

- **Videogames.** As mentioned in the section 3.1.2, the three major companies in the video game industry had each just released a game at the end of January, so there would be quite a few posts on social media talking about both companies and their recent releases.
- **Football.** The winter transfer window in European football closed on 31 January. In the English league, practically all of the most important clubs in the country -those that make up the so-called "Big Six"- had made significant additions or deletions to their squads, which is why fans have expressed their opinions regarding these movements on social networks, generating a significant amount of data to analyze. In this case, each of the teams would be a company, as they have their own official Twitter account, and each player would be their respective "product to analyze".

Once the two possible topics on which the study could be conducted had been defined, information was obtained on the official profiles of each company, as well as the publications of each account in the last week and the tweets that mentioned the company and/or its products.

As an example, let us take a look at **how to obtain tweets that mention the videogame Fire Emblem Engage**, the latest release of the Nintendo company. To do this, a request is made specifying that the game be mentioned, either by the full name of the game or by using the acronym of the franchise (FE). Furthermore, we will limit the results to those tweets that are neither replies nor retweets of any kind (simple or mentions), and have been published before 23:00 on 7 February 2023. In 4.2 we see the implementation of this from the previously established configuration, obtaining up to 50,000 tweets that meet these conditions. The requests are chained together using the tokens present in the metadata of the given response. This makes it possible to identify whether there are still results that can be extracted or, if not, the chain of requests must cease. In addition, a 1 second hibernation is included after each request to avoid exceeding rate limits.

```
q = '((fe engage) OR (fire emblem engage)) lang:en -is:retweet -is:reply -is:quote'
# First query
datos = client.search_recent_tweets(query=q, end_time="2023-02-07T23:00:00Z", tweet_fields=tweetFields, user_fields=userFields, place_fields=placeFields, media_fields=mediaFields, next_token=None, max_results=maxResults)
for i in range(pagTimes-1):
    tweets = client.search_recent_tweets(query=q, end_time="2023-02-07T23:00:00Z", tweet_fields=tweetFields, user_fields=userFields, place_fields=placeFields, media_fields=mediaFields, next_token=datos['meta']['next_token'], max_results=maxResults)
    # Aggregate collected data
    if ('data' in tweets): # Check whether the petition has been successful
        for tweet in tweets['data']:
            datos['data'].append(tweet)
    else:
        break
    # Update metadata for the following request
    datos['meta']['oldest_id'] = tweets['meta']['oldest_id']
    datos['meta']['result_count'] += tweets['meta']['result_count']
    if ('next_token' in tweets['meta']): # Comprobar si quedan tweets que se puedan extraer
        datos['meta']['next_token'] = tweets['meta']['next_token']
    else:
        break
    print(datos['meta']['next_token'])
    # Avoid exceeding rate limits
    time.sleep(1)
```

Code 4.2: Example of getting the tweets related to a specified query

4.2 Web scraping

Web scraping refers to the process of automatically extracting information from a web page, either by using the HTTP protocol or by using a web browser. Although this process can be carried out manually by the person who wishes to obtain the data, it is usual to use bots or spiders that carry out a systematic search on the Web to find the desired information. This methodology is followed by search engines in order to provide their users with the requested results.

In the field of data science, web scraping allows access to applications that do not offer an API to facilitate access to the information they contain, making them a vital means of obtaining data. In general, the methodologies used to carry out web scraping can be grouped into six categories:

- **Human extraction.** The most basic procedure, consisting of a person manually analyzing and extracting information from a website. Although very inefficient and unfeasible for large-scale projects, it may sometimes be the only way to obtain information from sites that use methods to prevent automated extraction bots.

- **Text patterns.** Search from a prefixed plain text match (such as using the *grep* command in Linux) or given regular expressions.

- **DOM analysis.** Nowadays, most Internet sites are **dynamic web pages**; see, they are based on an application that modifies the basic structure of the DOM (Document Object Model) of the web to dynamically display the content of the site. This results in a highly hierarchical formatting of the pages, which makes it possible to easily retrieve the content of the pages based on this structure.

- **Semantic annotation.** Pages may include metadata or other kinds of semantic annotations that provide additional information about their structure. If they are included in the pages themselves, this would be a special kind of DOM analysis. However, sometimes such information is stored in a separate semantic layer, which allows for pre-analysis before data extraction.

- **Computer Vision.** The most recent approaches in this field seek to apply Artificial Intelligence and Computer Vision techniques to automate human extraction using these tools.

For this project, it has been decided to implement a Jupyter notebook that obtains information from a website based on the DOM structure of the website. The most common way to do this is through the use of parsers such as **Beautiful Soup** [10], which allows its users to browse, search and modify the structures of files written in mark-up languages such as HTML or XML.

4.2.1 Metacritic

Metacritic is a website that compiles reviews of films, television programmes, music albums, videogames and, formerly, books. For each product, a weighted average of the scores of each review is made, which are colour-coded (green, yellow and red) for visual impact.

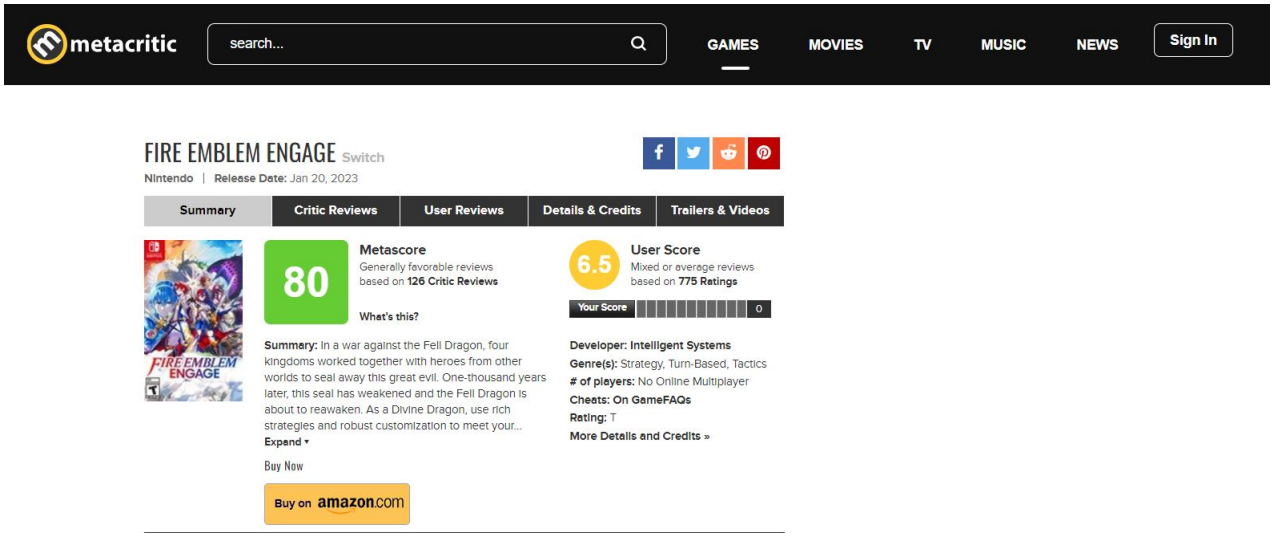

Figure 4.2: Example of a videogame overview page

Launched in 2001, its idea was to collect a wider range of ratings across a variety of media than Rotten Tomatoes, which was dedicated solely to film reviews. Nowadays, it has won two Webby awards for excellence as an aggregation website and is considered the leading online site for collecting videogame reviews, with many companies using it as a yardstick when planning their projects and releases [59]. However, this kind of information sources should be examined with caution, as they can sometimes suffer from the phenomenon known as **review bombing**, which is a massive influence of user reviews -whether positive or negative- to affect in some way the popularity or reputation of a product, service or company [62].

Although there are unofficial APIs for extracting data from Metacritic, due to the didactic nature of this work, it has been considered convenient to implement a notebook that exemplifies how to extract data from a website, as the methodology used can be extrapolated to any other web scraping project and it is sufficient to adapt the methods used to the format of the website to be analyzed.

4.2.2 Selected case study

Despite the fact that in the section 4.1.3 two different databases were obtained, as discussed in section 3.1.2, the study will focus on **videogame companies and their recent releases**, as the existence of review sites such as Metacritic provide additional information for analysis. The three games chosen have their own Metacritic pages that compile reviews from the trade press and users. However, in the case of Forspoken and Hi-Fi Rush, there is an additional page for each one because both titles were also released on PC and not only on the next generation consoles of each of the companies. Therefore, we will also collect the reviews for the PC releases.

For each video game, three data files will be sought to collect the general information available on the site: summary of the total number of existing reviews, reviews by specialised critics and user reviews. As an example, we will explain how the process of obtaining the latter was carried out. To do this, the DOM of the page had to be analysed. Once scrutinised, it was identified that the structure of the page is organised on the basis of a hierarchy of * * which are endowed with a particular class to identify them. Specifically, the reviews were contained within a list arranged as individual items following the scheme in the figure below 4.3.

Slight variations in structure, such as the body of the review, should also be considered when implementing it. Some users write reviews with very long texts, so that the page creates two **; one with a preview of the text that can be enlarged to the second by clicking on the ellipses, so the full review is located at **.

In addition, due to the large number of user ratings, reviews can be sorted according to various criteria such as date of age or rating. The default criterion is based on the usefulness of the review as rated by the community. To access the rest of the reviews, you only have to change the requested page by adding the *query* "?page=n", where n is the page number. In case there were no more ratings, instead of the ordered list, Metacritic displays the text *"There are no user reviews yet - Be first to review {Nombre del juego}."*, which will indicate when the extraction process should be completed.

A more in-depth analysis of the total data obtained, including an overall summary of all existing reviews, as well as reviews in the trade press, can be found in the section 4.3.2.

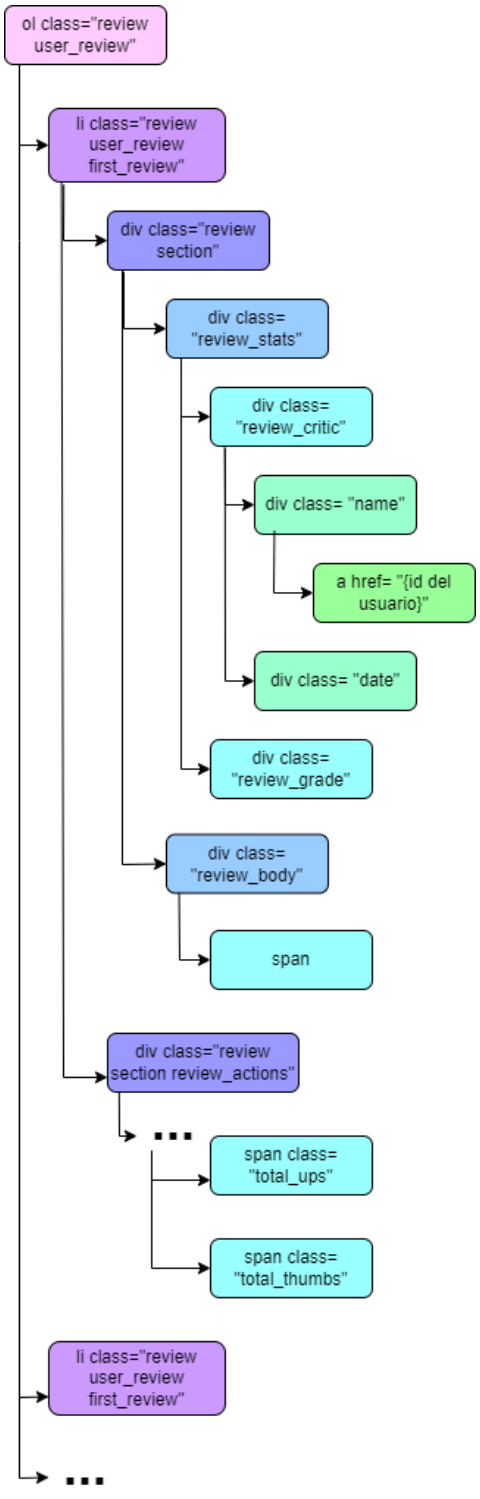

Figure 4.3: Outline of the DOM of user reviews

4.3 Data structure

Before being able to perform any kind of operation with the data, it will be crucial to make a first analysis of the structure and format of the data, in order to be able to understand what information we have and how we can make use of it. In the case of the data obtained by web scraping, we have already stored it in the corresponding Jupyter notebook, defining a structure that is beneficial to us, although a second analysis of the data is never too much. For the data obtained from Twitter, a somewhat more thorough scrutiny will be appropriate, as a large amount of data was obtained, each with a large number of fields and attributes.

4.3.1 Twitter

The raw data obtained from Twitter is structured in three folders, each containing a file whose name corresponds to one of the companies or one of the titles to be analyzed. These are json files composed of dictionaries, which consist of attributes and values, or other dictionaries that allow the content of the data to be nested.

- *users.* General information on the official profiles of the companies. It includes the unique profile identifier (id), as well as general statistics such as the number of followers or the total number of tweets made by the user.

```
data :  <class 'dict'>
   username :  <class 'str'>
   name :  <class 'str'>
   created_at :  <class 'str'>
   verified :  <class 'bool'>
   id :  <class 'str'>
   description :  <class 'str'>
   public_metrics :  <class 'dict'>
      followers_count :  <class 'int'>
      following_count :  <class 'int'>
      tweet_count :  <class 'int'>
      listed_count :  <class 'int'>
```

Figure 4.4: User data and format

- *tweets.* Tweets that include the keywords indicated in the request. The information is contained within the dictionary "*data*", as "*meta*" is only the metadata used during the API request chaining, although it includes a total of the results obtained; "*result_ count*". Each tweet consists of the following fields, which can be found at [72]:

- *author_id*. Unique identifier of the user who made the publication.
- *entities*. The entities provide metadata and additional contextual information about the content published on Twitter. They are JSON objects that provide additional information about hashtags, urls, user mentions... associated with a Tweet.
- *context_annotations*. Semantic annotations that the platform makes on the tweet based on keywords, hashtags or mentions relevant to a given topic.
- *public_metrics*. Public engagement metrics (retweets, replies, likes...) of the Tweet at the time of the request.
- *lang*. Language of the publication.
- *id*. Unique identifier of the tweet.
- *edit_history_tweet_ids*. Unique identifiers indicating all versions of a Tweet. For Tweets with no edits, there will be one ID. For Tweets with a history of edits, there will be multiple IDs, arranged in ascending order reflecting the order of edits. The most recent version is the last position in the array.
- *created_at*. Time of posting the tweet.
- *reply_settings*. Shows who can reply to the tweet. The possible options are "*everyone*", "*mentioned_users*" and "*followers*".
- *conversation_id*. ID of the original tweet of the conversation, if it is a thread. If it is an individual post without any kind of interaction, it corresponds to the tweet ID itself.
- *possibly_sensitive*. Indicative of whether the content of the tweet is potentially sensitive, either because the creator has marked it as such, or because a Twitter moderator has deemed it so.
- *text*. Tweet content (text) in UTF-8 format.

In total, both Playstation and Xbox got 50,000 tweets, while Nintendo only got 40,000. This difference is due to the fact that Nintendo is a Japanese-based company with a special emphasis on its domestic market, so, despite being highly relevant, it does not enjoy such a strong impact in the West as its competitors. For each title, a total of approximately 7,000 tweets have been obtained for each game.

- *usersTl*. Tweets that the user has made or retweeted. Its format is similar to the previous one, although this folder only consists of three files corresponding to each of the profiles of each company. The number of tweets for each company in the week in which the data was collected ranged from 2,300 for Nintendo to 3,000 for Xbox.

The total data collected has a certain **consistency with respect to the set of data obtained for the English Premier League**, as between 40,000 and 50,000 tweets were obtained for each team; while for each player there were approximately 7,000 tweets, so that the two case studies considered would have been equally valid for the study. The most curious case is the transfer of the player Jorginho, who moved from one team in the Big Six to another (from Chelsea to Arsenal), so the number of tweets obtained that talk about him double the usual number (15,000).

4.3.2 Metacritic

The data obtained in the previous notebook were structured based on the dataframes of the pandas library. **Pandas** is a software library for data manipulation and analysis for the Python programming language. In particular, it provides data structures and operations for manipulating numerical tables and time series. The name derives from the term "panel data", an econometric term for data that combines a time dimension with a cross-sectional dimension. It is implemented as an extension of the **NumPy** library, which allows manipulation of vectors and matrices in Python.

Pandas supports both **series** (indexed vectors) and, of course, **dataframes**. Dataframes are multidimensional arrays with labels for rows and columns, which can contain heterogeneous or even null data. The main virtue of pandas is that it allows to handle dataframes efficiently, providing a large number of methods and operations for their manipulation.

	source	link	date	grade	scoreType	text	upThumbs	totalThumbs	helpfulness
0	NagisaNeko	http://www.metacritic.com/user/NagisaNeko	Jan 24, 2023	7	Mixed	I admit that this work is better than the prev...	19	19	1.000000
1	Belonski	http://www.metacritic.com/user/Belonski	Jan 25, 2023	6	Mixed	One of the best Fire Emblems in technical aspe...	17	17	1.000000
2	avantic00	http://www.metacritic.com/user/avantic00	Feb 6, 2023	0	Negative	Combat was great and all, but the amount of cr...	16	17	0.941176
3	Faetori	http://www.metacritic.com/user/Faetori	Jan 25, 2023	5	Mixed	The gameplay itself and the technical portion ...	16	18	0.888889
4	SomethingSom	http://www.metacritic.com/user/SomethingSom	Jan 26, 2023	5	Mixed	Gameplay is fun. The story and characters are ...	15	17	0.882353
...	...	...	...	...	...	...	...	...	...
379	Aurok11	http://www.metacritic.com/user/Aurok11	Mar 1, 2023	10	Positive	Honestly, after reading user reviews of FE Eng...	0	3	0.000000
380	hyper06	http://www.metacritic.com/user/hyper06	Mar 5, 2023	9	Positive	Fire Emblem Engage is an amazing game and the ...	0	4	0.000000
381	PMG-Writer	http://www.metacritic.com/user/PMG-Writer	Mar 18, 2023	9	Positive	Nintendo and the developers from Intelligent S...	0	0	NaN
382	shw079	http://www.metacritic.com/user/shw079	Mar 20, 2023	8	Positive	This is my first fire emblem game and it is ov...	0	0	NaN
383	Snoopy64	http://www.metacritic.com/user/Snoopy64	Jan 31, 2023	7	Mixed	The gameplay alone deserves a 10, but the othe...	0	0	NaN

Figure 4.5: Dataframe with user reviews of Fire Emblem Engage

The flexibility of this format allows structuring both Metacritic and Twitter data, as dataframes can be created by importing csv or json files, as the conversion is based on simply using the *read_csv()* or *DataFrame()* methods of the pandas library, respectively.

Turning to the Metacritic data itself, this has been organized into three folders, which generally contain five files, one for each game and platform it is available on (as the PC ratings for Hi-Fi Rush and Forspoken are also included).

- *overviews.* Summary of all reviews for each game. Average total score and number of reviews, differentiating whether they are positive, negative or in-between. Two entries are included, one for all user reviews and one for trade press reviews. It should be noted that the totals obtained will not coincide with the number of reviews present in the following two folders, as both critics and users have the option to rate the product numerically without the need to provide a comment to accompany their rating.

- *criticReviews.* Within this folder, a distinction is made between reviews that include a numerical rating (*scored*) and those that do not (*unscored*). In the case of reviews with a rating, there are six columns of data:

  - *source*. Name of the entity author of the review.
  - *link*. Link to the original review of the game on the author's website, as Metacritic only includes a general summary of the review.
  - *date*. Date on which the review was published.
  - *grade*. Score given to the game out of 100.
  - *scoreType*. Type of rating given by Metacritic based on the score given. If it is lower than 50, it is considered negative, while a score equal to or higher than 75 is considered positive. If the rating is in the range between the two values, the review is rated as intermediate or mixed. This scale is only applicable to videogame reviews by trade press, as both critics and other artistic media have different ratings.
  - *text*. Text of the review itself.

  Reviews without score follow the same structure, but omitting the columns *grade* and *scoreType*.

- *userReviews.* A collection of all the reviews made by users. It follows a structure analogous to that of the trade press reviews, as can be seen at 4.5.

  - *source*. Name of the user writing the review.
  - *link*. Link to the user's profile.
  - *date*. Date on which the review was published.
  - *grade*. Score given to the game out of 10.
  - *scoreType*. Type of rating given by Metacritic based on the score given. In this case, it is considered positive if it is higher than 7, negative if it is lower than 5 and intermediate in any other case.

- *text*. Text of the review itself, which can be as long as desired, as it is not simply a summary of the review.
- *upThumbs*. Number of users who found the review useful.
- *totalThumbs*. Total number of users who have given their opinion on the usefulness of the review.
- *helpfulness*. Ratio calculated on the basis of the number of users who have considered the rating useful with respect to the total number of people who have given their opinion (value between 0 and 1).

Moreover, as seen in the aforementioned Pellarolo study [52], it should be noted that user reviews tend to give a lower rating than that given by specialist reviewers (see 3.4).

Chapter 5

Data preprocessing

Data collection is only the beginning of any data science project. Once saved, it is vital to know how to identify which data are relevant to the study, which are not useful in the context of the work or even those that may contaminate the development of the study. After this initial cleaning, and given that we are dealing with a project focused on the use of NLP tools, it is crucial to pre-process the data so that they acquire a format that facilitates their manipulation, as well as the visualization of the final results.

This final goal is what is known in NLP as a corpus. A **corpus** refers to a collection of authenticated text or audio, organized as datasets. In this context, the term "authenticated" indicates that the source of such information is a person with some fluency in the language. Once obtained, the corpus can be used for more complex tasks such as training AIs or Machine Learning models [35].

Most of these tasks will make use of Natural Language Toolkit or NLTK [13], a set of Python libraries and programs that allow a wide variety of NLP-related tasks to be carried out. These resources are developed thanks to the more than 50 corpora (plural of corpus) to which it has access, some of them as relevant as **WordNet**, a large database of English lexicons.

For this reason, this chapter will focus on keeping only the data that will be used throughout the study and formatting it in a suitable format for each of the phases of the data processing. The main steps of a data cleaning process will be visited, focusing on the specifics of this study and applying them to the data collected from both Metacritic and Twitter. As we want to perform sentiment analysis and topic modelling, the pre-processing will focus only on the textual information obtained from both ways (the text of reviews and tweets).

5.1 Data cleaning

The first step in processing data is the removal of all unwanted entities and characters. For this work, the following elements were identified for removal:

- Non-ASCII characters (non-printable control characters)
- Hyperlinks and links
- HTML tags
- Twitter entities (hastags, urls and usernames)
- Punctuation marks

In addition to removing all these elements, all words will also be unified to lower case to avoid redundancy of terms, as otherwise the words "Nintendo" and "nintendo" could come to be seen as distinct even though they both refer to the Japanese company. This process is known as **case folding**.

The recognition of all these entities has been done thanks to the use of the NLTK library mentioned above, together with the use of **regular expressions** (sequence of characters specifying a matching pattern), which can be manipulated thanks to the Python module **re**.

5.2 Language filtering

Once the initial processing of the data has been carried out, it will be necessary to keep only the texts written in English. While such sifting was already done when making requests for tweets using language filters, for Metacritic reviews no such filtering took place, so it is mandatory to discard those texts that are not written in the desired language. The necessity of this step becomes evident when looking at, for example, the reviews of Hi-Fi Rush for Xbox, because, as can be seen in figure 5.1, apart from the English reviews there are also reviews in Portuguese, Turkish and even Chinese.

Language detection is done thanks to the **langdetect** library, which is a direct adaptation of the language detection mechanism implemented by Google for Java.

```
0       Amazing Game, one of the best surprises ever! ...
1       It's highly recommended and one of the best vi...
2       Jogo perfeito! Sua jogabilidade, gráficos e hi...
3       The good:\rAn interesting mix of Sunset Overdr...
4       Başında oturdun mu saatlerce kalkamıyorsun. Yı...
                              ...
1419    Love it. Great gameplay. Incredibly immersive....
1420    Worst game, the worst game in my entre life pl...
1421    This game lives up to the hype. The world conc...
1422                      我歌唱火焰，在我的眼睛周围，他们永远不会害怕，就像敌人奔
向太阳
1423    Amazing game! I loved the art style, all plent...
Name: text, Length: 1424, dtype: object
```

Figure 5.1: User reviews of Hi-Fi Rush on Xbox

5.3 Tokenization

Text tokenization, commonly abbreviated just as **tokenization**, is the process of dividing text into simple units called tokens. For most languages, such a division is made on the basis of whitespaces or punctuation marks, although for languages that do not include spaces, such as Chinese or Japanese, the partitioning requires greater complexity. However, such a rigid division into units risks losing some of the meaning of the original text.

However, the process used in this section has been greatly simplified, as the consideration of joint meanings is done in the sections dealing with data modelling and analysis, as explained in sections such as 6.3. Moreover, as all the texts analysed are written in English, the separation into tokens can be done simply by words.

The practical implementation of this section has been carried out thanks to the tool **TweetTokenizer** provided by NLTK, specially designed for the tokenization of texts identified as tweets. It has also been used in the case of reviews, both being short texts where an idea or opinion is expressed. Another alternative for tokenization would be the **gensim** library, designed for text modelling, document indexing or identifying similarities between corpora, which in turn includes tools for text tokenization [54].

5.4 Stop words

In 1959, Hans Peter Luhn proposed that the most frequent words in a text are not the ones that provide the most information. This has been endorsed by several subsequent studies such as [36], where it can be seen that the most relevant terms are those that appear with an intermediate frequency, and not the words that appear most frequently or hardly at all. In fact, the most frequently occurring terms can even be deleted from a text without it losing its primary meaning.

Based on this idea, it is Luns himself who first coined the terminology **stop words** to refer to those terms lacking a meaning of their own -at least not by themselves- but which are used very frequently in most languages. Some of these stop words are articles, pronouns, propositions or conjunctions.

NLTK provides a dictionary of stop words in several languages so that they can be easily removed from any corpus, simplifying the task for data scientists.

5.5 Stemming and lemmatization

Word normalization consists of unifying words or tokens around a common standard format. This process usually starts by performing the **case folding** phase already discussed in the section 5.1, and then obtaining the underlying meaning behind each word, which can be retrieved thanks to their roots. The retrieval of these can be done in two different ways:

- **Stemming.** Reduction of inflected or derived words to their root. Thus, inflections derived from variations in gender and number can be unified under the common root they share, only by eliminating the corresponding desinence. This crude procedure can lead to errors in the final standardized text, such as incorrect meanings or misspellings, as shown in the figure 5.1.
- **Lemmatization.** Refinement of the previous method, consisting of grouping the inflected forms of a word so that they can be analyzed as a single element, thanks to the dictionary form of each word.

Thus, stemming is a simple process of truncation (and sometimes replacement by a general desinence) for obtaining the root of words; whereas lemmatization involves the use of more sophisticated methods of full morphological analysis of the word, distinguishing whether a desinence or an affix (a morpheme that only modifies the root grammatically or provides some additional meaning) has been added to the root. Both procedures can be performed by NLTK functionalities, such as *SnowballStemmer* for stemming or *WordNetLemmatizer* for lemmatization.

```
# Example of stemming errors
print('Pre-processed and tokenized text: ',tweets['Nintendo'].
    finalText[36833])
print('Stemmed text: ',tweets['Nintendo'].stemmedText[36833])
print('Lemmatized text: ',tweets['Nintendo'].lemmatizedText[36833])
```

```
Pre-processed and tokenized text:  february means definitely
    official nintendo direct watch 2019 february 13th 2020 march 26
    th 2021 february 17th 2022 february 9th nothing ever 100
    guaranteed odds good
Stemmed text:  februari mean definit offici nintendo direct watch
    2019 februari 13th 2020 march 26th 2021 februari 17th 2022
    februari 9th noth ever 100 guarante odd good
Lemmatized text:  february mean definitely official nintendo direct
     watch 2019 february 13th 2020 march 26th 2021 february 17th
    2022 february 9th nothing ever 100 guaranteed odds good
```

Code 5.1: Difference between stemming and lemmatization.

Chapter 6

Topic modelling

Topic modelling is a machine learning technique, from the branch of natural language processing, that tries to assign topics to a set of texts. Specifically, it refers to statistical models that allows the discovery of underlying topics in collections of texts of any kind.

Topic identification models are essentially iterative algorithms that work with **document feature matrices** (matrices describing the frequency of occurrence of terms across a series of documents) to group them based on their common elements. While matrices often simply capture the frequency of occurrence of each word, they can also refer to nouns or entities by name. A simple example of application would be a collection of documents where words such as “match”, "team" or “score” appear, which could be grouped under a single topic called "sport"; while other terms such as “case”, “law” or “crime”, which also appear in the documents, would be grouped under the topic of “legality” [17]. Figure 6.1 illustrates the case described by considering the analogous English terms.

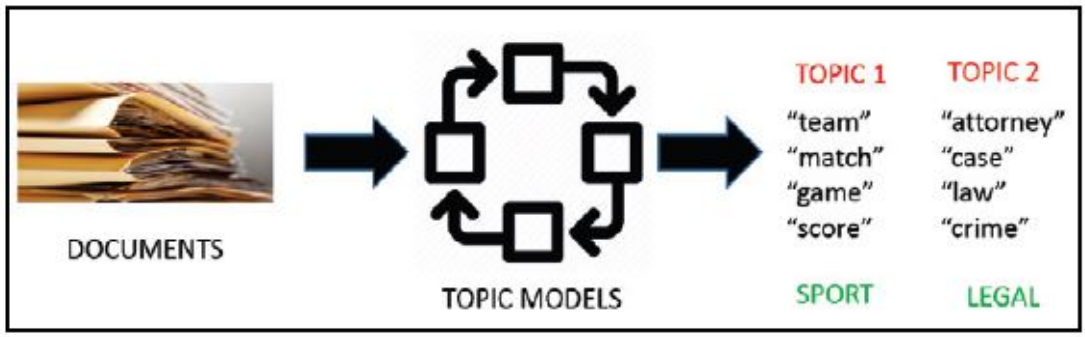

Figure 6.1: Basic operation scheme of a topic modelling process

Throughout this chapter, a preliminary analysis of the data will be carried out before applying models for topic extraction. In particular, two specific topic modelling models will be applied: Latent Dirichlet Allocation and Biterm Topic Modelling.

6.1 Exploratory Data Analysis

In the previous chapters, an analysis of the structure of the data obtained has been carried out, as well as a processing of the data. However, the content of the data as such has hardly been examined. Although the figure 2.1 explained the basic steps of a data science project, a real project does not follow a linear methodology, but at the end of each development phase, the new results obtained are used to improve the previous steps. Therefore, a truer approximation of these steps would be the one shown in the figure 6.2.

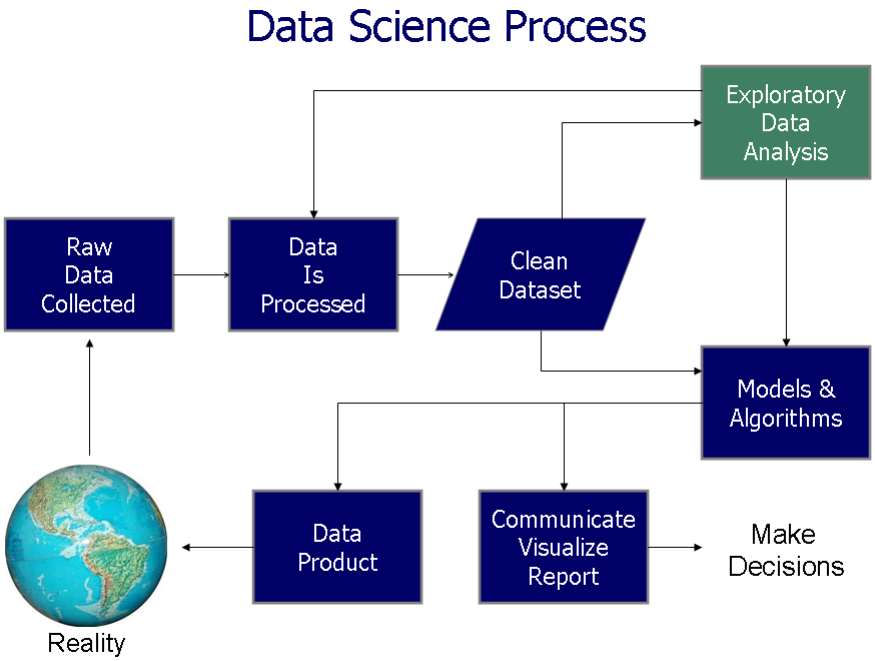

Figure 6.2: Actual outline of the development of a data science project

Exploratory Data Analysis (EDA) refers to the key process of conducting initial investigations of data to uncover patterns, detect anomalies, test hypotheses and test assumptions with the help of summary statistics and graphical representations [51]. This allows us to determine the best way to manipulate the data we have in order to obtain the answers we are looking for, as well as to corroborate the appropriateness of the tools we have already used.

Originally developed by the American mathematician John Turkey in his book of the same name published in 1977, today EDA strategies can be classified into four categories depending on the type of data (univariate or multivariate) considered and the representation of the data (graphical or not) [75].

- **Non-graphical univariate.** Basic description of the data to find patterns in the data. As it is a single variable, it is not necessary to study causes or relationships with other variables.
- **Graphical univariate.** Variant of the previous method that seeks to provide a clearer view of the data through the use of histograms or box plots.
- **Non-graphical multivariate.** In this case it is necessary to study the dependence and correlation of the variables. The quickest way to do this is by cross-tabulation or statistics such as the correlation coefficient.
- **Graphical multivariate.** The most effective way to visualize the relationship between several variables is through the use of graphs such as heat maps or bubble charts.

6.1.1 WordCloud

A **Word Cloud** is a way of visually representing a text based on the terms that appear most frequently in it, increasing in size depending on whether they are frequently repeated throughout the text. Therefore, it would be framed as a graphical univariate EDA method. In this way, it is a very visual way to start a process of topic modelling, as it allows to identify those topics that appear more frequently throughout the text.

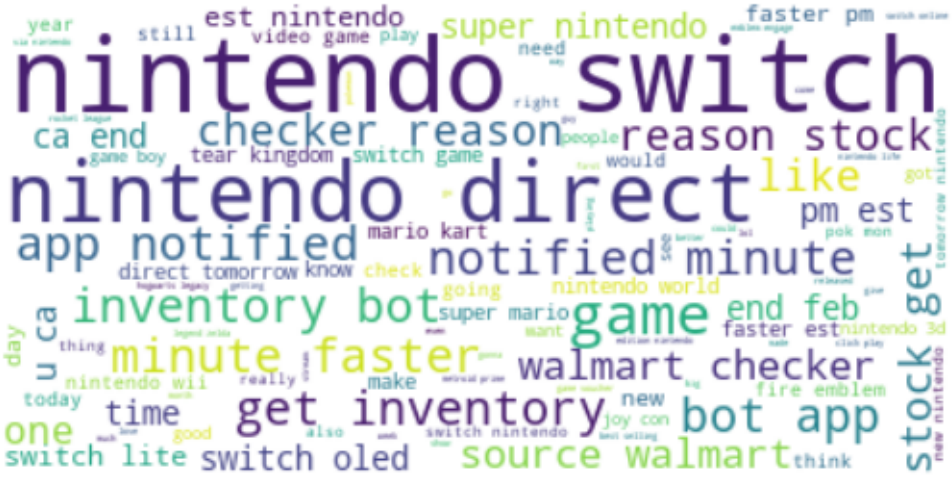

Figure 6.3: Word cloud of Nintendo's tweets

For this project, we have used the Python implementation **WordCloud** present in [45], which allows to intuitively generate word clouds giving extra customization options such as limiting the maximum number of terms that appear or specifying the set of stop words that should not be considered.

[74] is a great introduction to getting to grips with all the tools and methods available in the library. For example, a key functionality is the ability to decide whether sets of two terms can be considered rather than all being considered individually. This is key for identifying terms such as the names of consoles, as seen with the Nintendo Switch example in the figure 6.3. This idea is the basis of the Biterm model developed in 6.3, and can be extended to even more terms as, for example, Fire Emblem Engage is three words and the model only identifies two of these.

6.1.2 Results

When generating the word clouds, errors were detected during the cleaning process, such as the fact that some stop words were not discarded as the subject of sentences, or that apostrophized verbs remained as such. This made it necessary to go back to the development of the data processing steps until valid results were obtained, which exemplifies the circular nature of the project's development already discussed in the figure below 6.2.

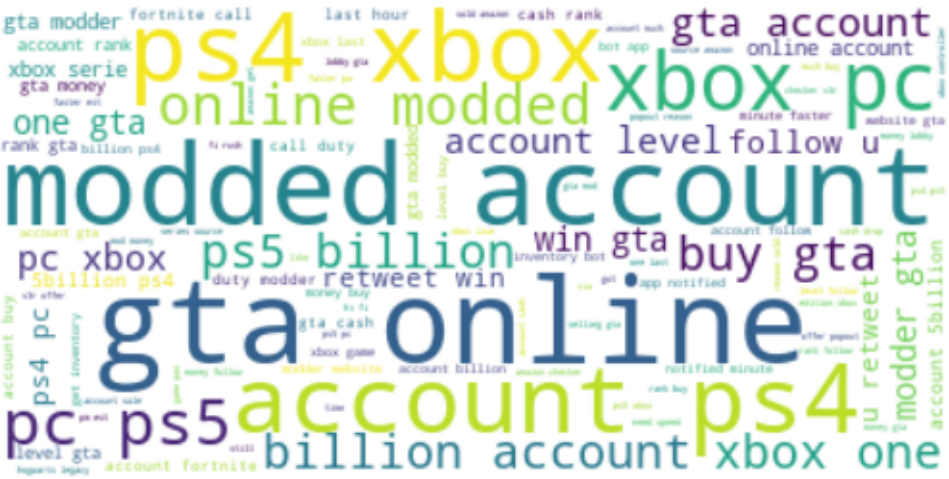

Figure 6.4: Word cloud of Xbox's tweets

With regard to the word clouds themselves, this first analysis makes it possible to identify some elements that will become clearer in later phases of the analysis process:

- **Nintendo**. When considering the tweets that mention the Japanese company, we can see in the figure 6.3 some recurring themes such as Nintendo Direct, an event that the company organises every so often to present its new products and usually announces shortly before it takes place; or Walmart, which had just denied a series of rumours that spoke of a possible imminent launch of Advance Wars, another Nintendo video game saga [23].

- **Xbox**. Among the tweets mentioning the US company, a predominance of terms such as gta or modified accounts can be seen in the figure 6.4. This is because the company Rockstar had just released a patch for the PC version of its game GTA online, due to vulnerabilities in the software that allowed hackers to modify the statistics of other players, causing the game itself to expel users who had been targeted by hackers when they were detected as having illegitimately modified their statistics [55]. Although this modification only occurred on PC, the massive appearance of these terms when searching on Xbox shows the close relationship Microsoft has established between the two markets.

- **Forspoken**. While reviews may not be as high a target for this kind of analysis as tweets, as they have a more specific and concrete focus, some initial insights can still be gained, as in the case of the figure 6.5. As might be expected, topics that speak to the character of the game itself are mentioned, such as *open world*, *magic* or *combat*, although terms such as *bad* or *boring* appear, foreshadowing the results that will be obtained when carrying out the sentiment analysis of the section.

Figure 6.5: Word cloud of Forspoken's reviews on PS5

Only some of the word cloud generated have been commented on, highlighting those of tweets with respect to those of reviews, as the latter have a greater amount of data to scrutinize. To see and analyze all of them in depth, just refer to the corresponding Jupyter notebook.

6.2 Latent Dirichlet Allocation

LDA is a probabilistic generative model of a corpus. The fundamental idea behind it is that documents contain several underlying topics, which are characterized by the distribution with respect to the existing words. In essence, we could define it as an unsupervised model within the branch of Machine Learning that allows the induction of terms relative to large sets of text. In order to understand how it works, it will be necessary to analyze the general functioning of the language models used in NLP, which are usually Bayesian (see section 7.2.1).

6.2.1 N-grams

If one considers the sentence "*I am going to park my car in*", it is common to assume that structures such as "*the garage*" or "*the parking*" will follow this information, rather than others such as "*the kitchen*" or "*the umbrella*". Although this may seem obvious, it motivates the study of the probability of occurrence of terms in NLP. In any Natural Language Processing task, the calculation of the probability of occurrence of a word is crucial, either to identify if there has been an error when transcribing a spoken speech or if a translation process is to be carried out.

Models that assign probabilities to sequences of words are called **language models**. The simplest of these is the **n-gram**, which is based on the application of the **probability chain rule**[1], in which it is sufficient to substitute random variables for the words that make up a sentence or document:

$$P(w_{1:n}) = P(w_1) \cdot P(w_2|w_1) \cdot ... \cdot P(w_n|w_{1:n-1}) = \prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

considering that we are evaluating a document, which is essentially a collection of words ordered as a sequence; $d = (w_1, w_2, ..., w_n)$ and $w_{1:k}$ is the sequence of the k first words. This expression shows the relation between the joint probability of the sequence of words with respect to the conditional probability that the previous words appear. However, since we do not know these prior conditional probabilities either, it is most common to resort to approximation methods. In particular, the Markov property is usually considered valid, in order to work with Markov models.

A **Markov process** refers to probabilistic models in which the probability of a future event depends only on those immediately preceding it. An example is Markov chains, in which the probability of the next event depends exclusively on the one immediately preceding it, which would allow estimating 2-grams (or **bigrams**) knowing the probability of occurrence of one of them. For the case of n-grams, the calculation of the conditional probability is based on the previous $n - 1$:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-N+1:n-1})$$

[1]The calculation of probabilities is always done in logarithmic format to avoid problems of **underflow** when multiplying numbers between 0 and 1.

Apart from this estimation, the calculation of the conditional probabilities is carried out using **maximum likelihood estimation methods**. For the case at hand, we can simply consider a large document serving as a corpus and estimate based on the number of occurrences of these structures, normalizing it to a range between 0 and 1:

$$P(w_n|w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}w_n)}{C(w_{n-N+1:n-1})}$$

being $C(w_{n-N+1:n-1w_n})$ the number of times the searched n-gram in question appears. This ratio is also known as **relative frequency**.

6.2.2 TF-IDF

However, these terms are not random variables as such, but words with associated meanings that condition their own probability of occurrence depending on the context. To study this there is **vector semantics**, which is the standard way to represent the meaning of words in NLP through their context. In mathematical terms, the underlying idea is to represent each word as a point in a multidimensional space that could be categorized as semantic, being drawn from the distributions of neighbouring words. The vectors for representing words are called **embeddings**, whose idea is similar to that of their mathematical analogue. A simple example of an embedding is the figure 6.6, which groups English words according to whether their connotation is positive, negative or neutral. These methods are particularly useful for topic detection or sentiment analysis.

Figure 6.6: Basic example of embedding for the detection of similar meanings

These distribution-based models use as their basis a matrix of co-occurrences, representing how often terms are linked. For a document, this is represented by the **term-document matrix**, where each row represents a word and each column a document in the corpus, considering the corpus as a collection of M documents; $D = \{d_1, d_2, ..., d_M\}$.

As discussed in the section 5.4, the terms that provide the most information are those that appear with an intermediate frequency [36]. In order to elucidate what these terms are, the **tf-idf algorithm** is used, which is based on the product of two terms:

- **Term frequency (tf)**. Frequency of occurrence of a word w throughout a document d; $\text{tf}_{w,d} = C(w,d)$. [2]
- **Inverse document frequency (idf)**. A Factor that gives more weight to terms that only appear in a few documents. For this, it is necessary to know the document frequency of a word w or **document frequency** ($textdf_w$), which indicates how many documents in the corpus it appears in. Thanks to this, the weight is calculated simply as the ratio $\text{idf}_w = \frac{M}{\text{df}_w}$.[3]

This therefore allows the relevance of each term to be calculated as the following product, which assigns a weight/importance between 0 and 1 to each word in a document:

$$\mathbf{w}_{w,d} = \text{tf}_{w,d} \times \text{idf}_w$$

The main drawback of this tool is that it provides almost no dimensional reduction of the analyzed terms and reveals little of the underlying relationships between documents. To overcome these problems, **Latent Semantic Analysis/Indexing** (LSA/LSI) has emerged, which essentially consists of applying a singular value decomposition on the resulting final matrix.

6.2.3 LDA

While LSA gives acceptable results, a new model has emerged in recent years that provides more accurate results for the identification of topics, and has proven to be more useful for clustering high dimensional datasets: Latent Dirichlet Allocation.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model, so it takes up the use of n-grams for the computation of probabilities. For this purpose, LDA assumes the following generative process for each document d_i in a corpus D:

1. Define $N \sim \text{Poisson}(\zeta)$.
2. Define $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n de $P(w_n|z_n,\beta)$, a multinomial conditional probability as a function of the chosen topic z_n.

[2] Other alternatives for this calculation are based on mitigating the effect of words appearing multiple times by taking the decimal logarithm; $\text{tf}_{w,d} = log_{10}(C(w,d)+1)$.

[3] It is also often mitigated by the use of the decimal logarithm; $\text{idf}_w = log_{10}(\frac{M}{\text{df}_w})$.

where the probability of the words is established thanks to a matrix $\beta \in \mathcal{M}_{k \times V}$ where $\beta_{ij} = P(w_j|z_i)$, assuming that the set of words is extracted from a vocabulary of size V and k is the dimension of the Dirichlet distribution, which determines the total number of possible subjects and has probability density:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_k^{\alpha_k - 1}$$

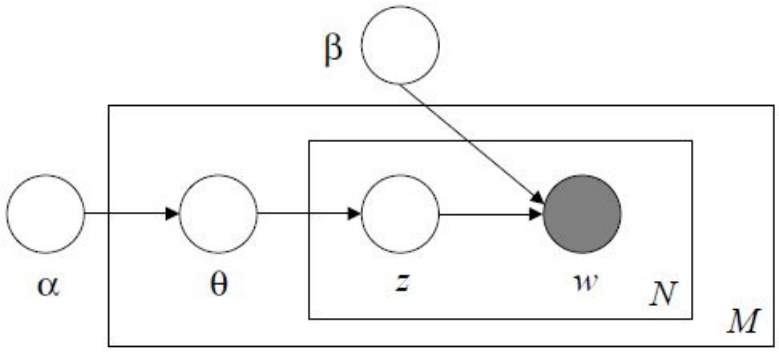

Figure 6.7: Graphical representation of the generation from LDA. The first table indicates that the process is carried out on the whole set of available documents, while the second does the same for each document (choice of topics and words).

In somewhat simpler terms, what is done is to assume Poisson and Dirichlet distributions over the models and then create the members of each group by maximizing the probability that a new word is a function of the current components of the group. The key to LDA is to consider the topics as **exchangeable random variables**[4], which makes it possible to give the probability of a set of words and topics such as:

$$P(\mathbf{w}, \mathbf{z}) = \int P(\theta) (\prod_{n=1}^{N} P(z_n|\theta) P(w_n|z_n)) d\theta$$

To obtain the word distribution, it is sufficient to consider the marginal distribution, obtaining the marginal distribution of each document as a mixture of weighted continuous distributions:

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) (\prod_{n=1}^{N} P(w_n|\theta, \beta)) d\theta$$

[4] A set of random variables $\{z_1, ..., z_N\}$ is called **exchangeable** if their joint distribution is invariant to permutations; $p(z_1, ..., z_N) = p(z_{\pi(1)}, ..., z_{\pi(N)})$. For infinite sets, the property is analogous if it holds for any finite subset of random variables.

6.2.4 Implementation and results

Having seen how LDA works theoretically, we can start using it to extract themes from the processed data. This will be done using **gensim**, already mentioned in the section 5.3, an open source library that contains a multitude of tools for unsupervised topic modelling and other NLP functionalities. In particular, it includes an LDA model called **LdaMulticore** for topic extraction, the results of which can be easily visualized as interactive workbooks, which can also be stored as html files. The model requires the following parameters to be specified:

- **id2word**. Relationship between all words and IDs assigned to them. This allows you to define the dictionary we will work with.
- **corpus**. Corpus of all documents already converted into a frequency matrix. This last step is performed thanks to the **doc2bow** method, which converts documents, which are assumed to be already tokenized and normalized, into tuples of id and number of token occurrences.
- **num_topics**. Number of topics to be extracted. Ideally, we would like to optimize the number of topics to be extracted, but because LDA is a model more focused on the analysis of long documents rather than short ones such as tweets or reviews, we will consider the number of topics for all models to be 10.

Once a model has been trained, its content can be visualized thanks to **pyLDAvis**, which provides an interactive format to easily understand the results obtained by LDA. The workbook is composed of two interrelated parts, as shown in the figure 6.8:

- **Intertopic Distance Map**. Bubble chart representing the different topics found in the text, representing the difference between them thanks to the distance calculated using multidimensional scaling (MDS). It allows the selection of specific topics, although all are grouped under a global one with numbering 0.
- **Most relevant terms**. Bar chart with the most relevant terms, which are decided on the basis of the relevance metrics indicated and can be adjusted using the parameter λ. By selecting a particular topic, we can see the specific frequency of the terms in the bubble in question. In addition, by selecting a particular term, the bubble chart is modified to adjust the size of the bubbles according to where the token appears most. An example of this can be seen in the figure 6.9, in which the topic 4 has been selected as well as the term *walmart*.

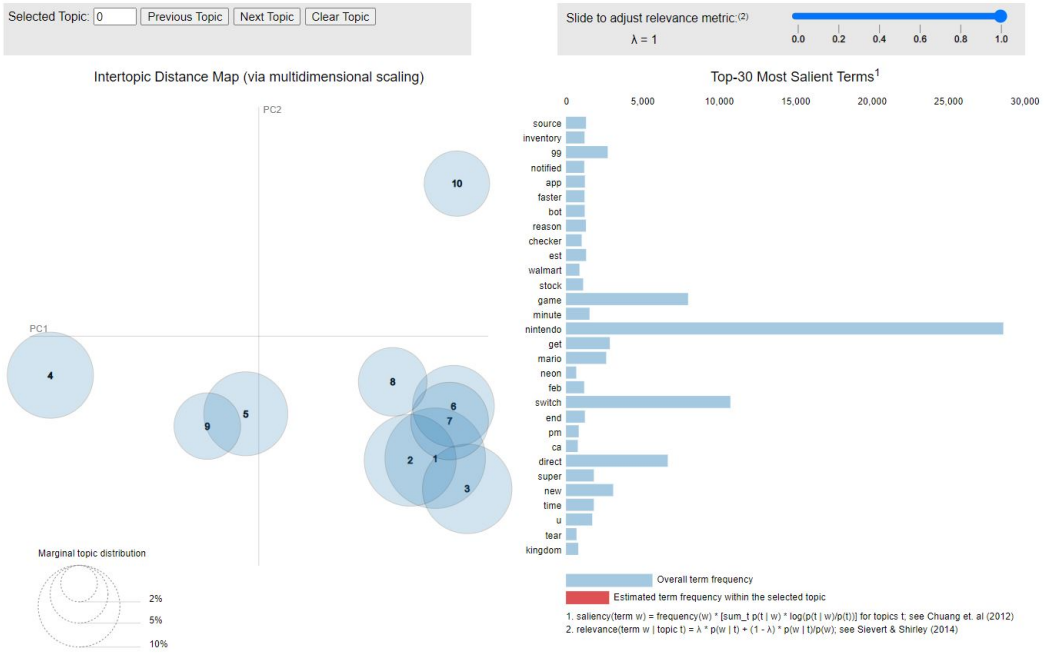

Figure 6.8: Visualisation of the model trained with tweets about Nintendo, considering 10 topics

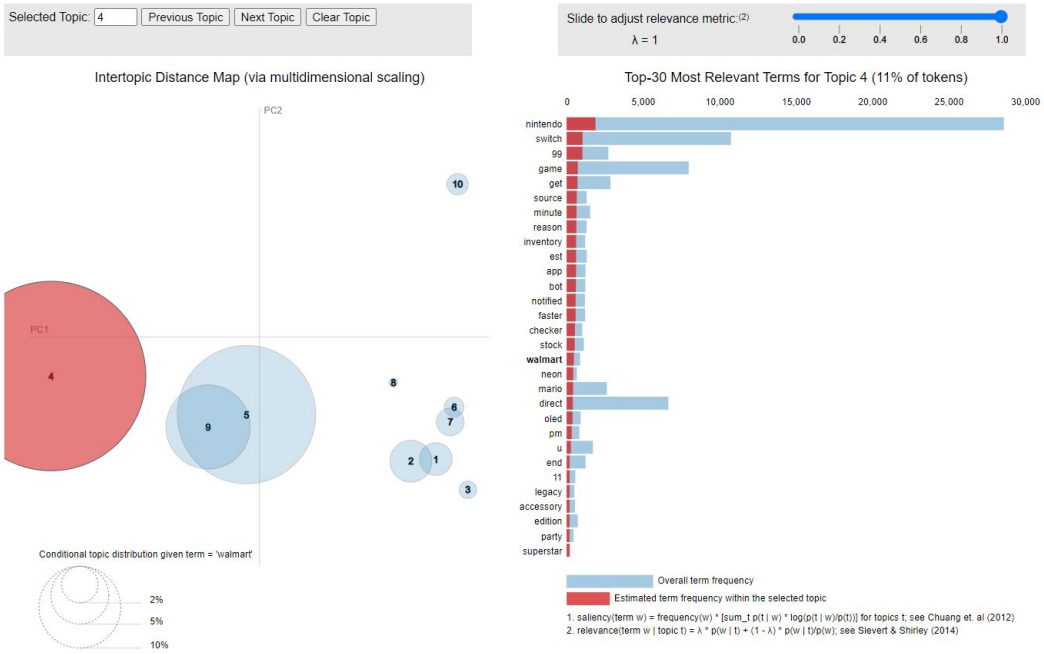

Figure 6.9: Nintendo model highlighting the fourth topic and the term *walmart*

Thanks to this, we can discover some of the hot topics in each of our datasets, although as there are a large number of them, here we will only make a few insights into these that can be extracted after careful scrutiny of the results obtained. With regard to the tweets, let us analyze which are the most commented topics for each of the companies:

- **Nintendo**. As already observed in the generated word cloud, tweets related to the Nintendo Direct are recurrent, which are located in the lower quadrant of the bubble chart; as well as reactions to the Walmart Canada announcement, which are located in the lower left quadrant (figure 6.9).
- **PlayStation**. Consulting the models reveals two major themes: Playstation Plus, which offered different content benefits for its subscribers in games such as Call of Duty Modern Warfare 2.0, Fortnite, or Apex Legends; and *broadcast*, due to tweets from gamers announcing that they are playing live on platforms such as Twitch.
- **Xbox**. The main highlight could already be observed during the exploratory data analysis. The predominant topic commented on by the community is the GTA update that prevented hackers from further modifying accounts. While its presence was already distinguishable in the generated word cloud, the clustering of all these terms into a single topic is particularly noticeable in the bubble graph generated by LDA, as can be seen in the figure 6.11. In addition, it can also be seen that other topics that are quite popular are online multiplayer games in vogue such as Fortnite or Call of Duty, as was the case on PlayStation.

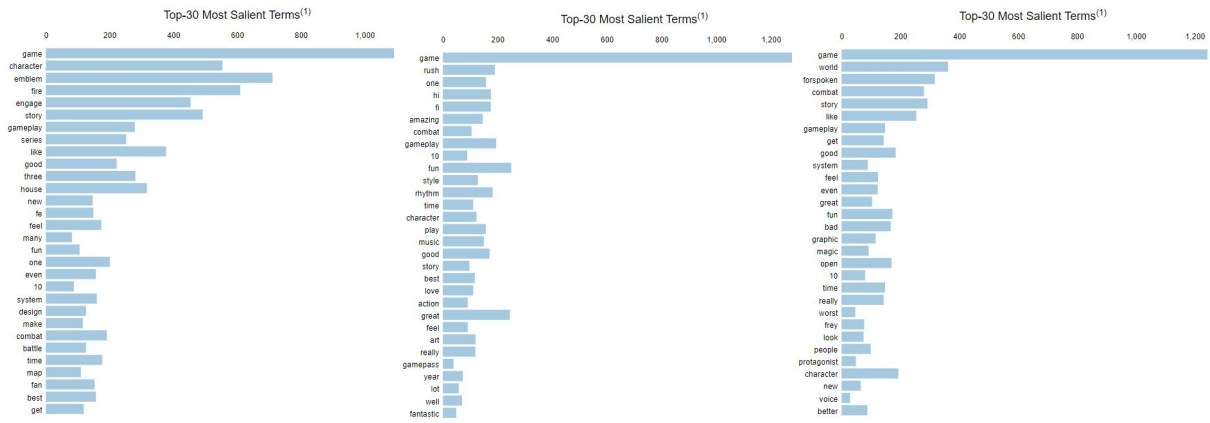

Figure 6.10: Most relevant terms in the reviews of each videogame on each company's consoles (FE Engage for Switch, Hi-Fi Rush for Xbox Series X and Forspoken for PS5; respectively)

As for the reviews of the different games, as they are more focused texts and without so much variability with respect to the subject matter they can address (as they are all evaluations of the same product), we will focus on the most relevant terms, to see if the results are coherent with those of the exploratory analysis of the data. As an example, let us look only at the reviews of each product related to the console belonging to each of the companies. As can be seen in the figure 6.10, both Fire Emblem Engage for Nintendo Switch and Hi-Fi Rush for Xbox X Series have a large number of positive terms associated with them, such as *best*, *good* or *fantastic*; in the former, talking about the combat (*combat*) or the design (*design*), while in the latter, terms such as *rhythm* or *music* stand out. On the other hand, **Forspoken for PS5** also has some of these positive terms associated with it, although others such as *bad* or *worst* are present, which seems to point to a certain negative reception, as was already evident in the word cloud created.

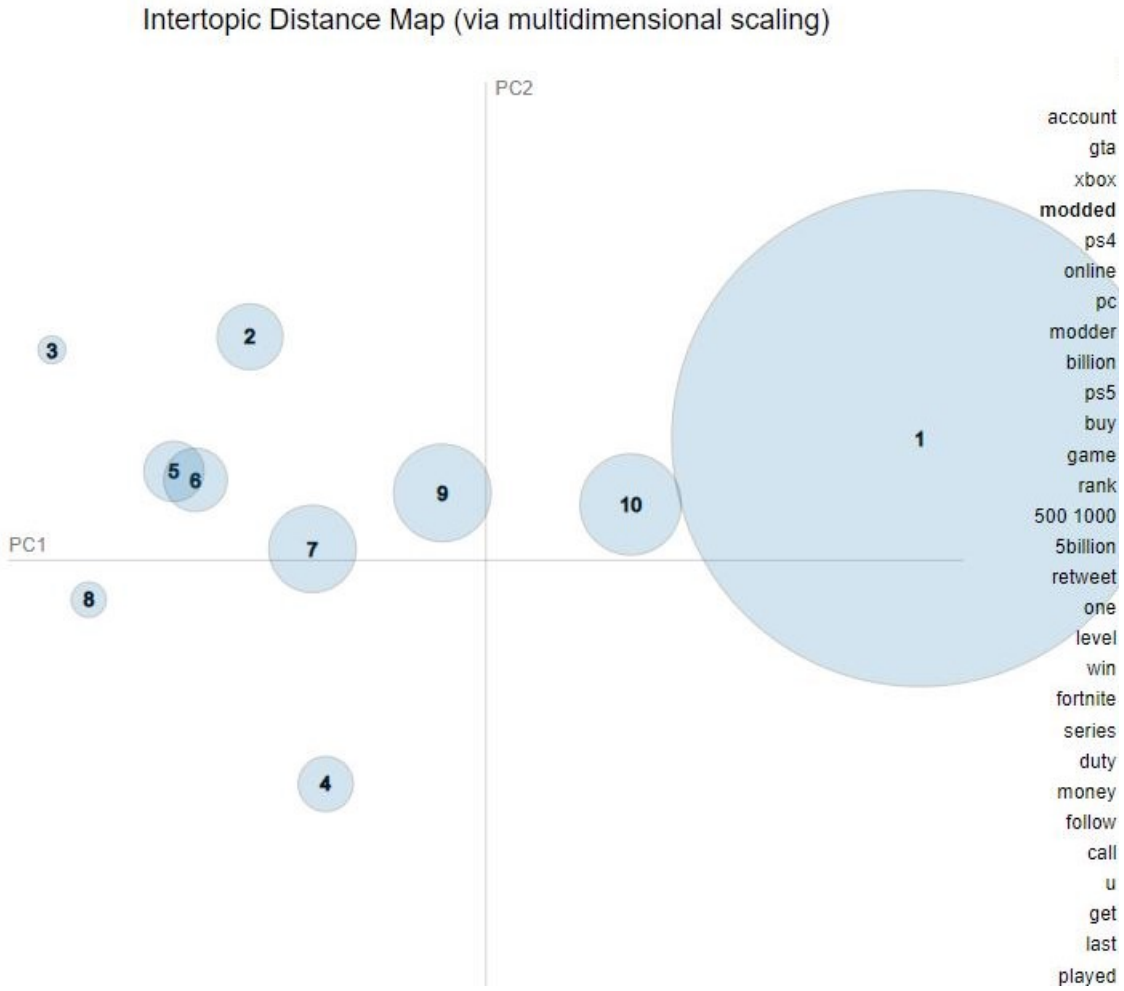

Figure 6.11: Xbox-related bubble chart by highlighting the term *modded*

6.3 Biterm Topic Model

While the results obtained so far have been favourable, it is true that the classical models already seen as LDA have some performance problems with short texts, such as tweets or reviews. The main reason is that conventional thematic models implicitly capture word co-occurrence patterns at the document level to reveal topics, which is why they are affected by the scarcity of data in short documents. To put an end to this problem arises the model known as Biterm, specifically designed for topic modelling in short texts [77].

Biterm Topic Model (BTM) uses the aggregated patterns of the whole corpus to learn topics and solve the problem of the paucity of document-level word co-occurrence patterns that conventional models have. The idea behind this is analogous to that of the 2-grams already seen, the only difference being that we consider pairs of co-occurring **unordered** words. The process of generating a BTM corpus is similar to that of LDA, but only using Dirichlet distributions:

1. For each topic z, choose a distribution of words by topic $\varphi_z \sim \text{Dir}(\beta)$.
2. Choose a topic distribution $\theta \sim \text{Dir}(\alpha)$ for the whole collection.
3. For each biterm b of the set of biterms B:
   (a) Choose a topic $z \sim \text{Multinomial}(\theta)$.
   (b) Choose two words $w_i, w_j \sim \text{Multinomial}(\varphi_z)$.

We see that the set of biterms B works for practical purposes like the dictionary V used in the LDA model, although here its elements are tuples $b = (w_i, w_j)$, and the joint probability of each biterm is calculated as:

$$P(b) = \sum_z P(z)P(w_i|z)P(w_j|z) = \sum_z \theta_z \varphi_{i|z}\varphi_{j|z}$$

Based on this, the extraction of topics from a document is based on the use of the following probability:

$$P(z|d) = \sum_b P(z|b)P(b|d) =$$

$$= \sum_b \frac{P(z)P(w_i|z)P(w_j|z)}{\sum_z P(z)P(w_i|z)P(w_j|z)} \frac{C(b,d)}{\sum_b C(b,d)} = \sum_b \frac{\theta_z \varphi_{i|z}\varphi_{j|z}}{\sum_z \theta_z \varphi_{i|z}\varphi_{j|z}} \frac{C(b,d)}{\sum_b C(b,d)}$$

whose calculation can be done by using the conditional probability rule and estimation from relative frequencies (analogously to the case seen for n-grams), where $C(b,d)$ is the frequency of occurrence of the biterm b in the document d.

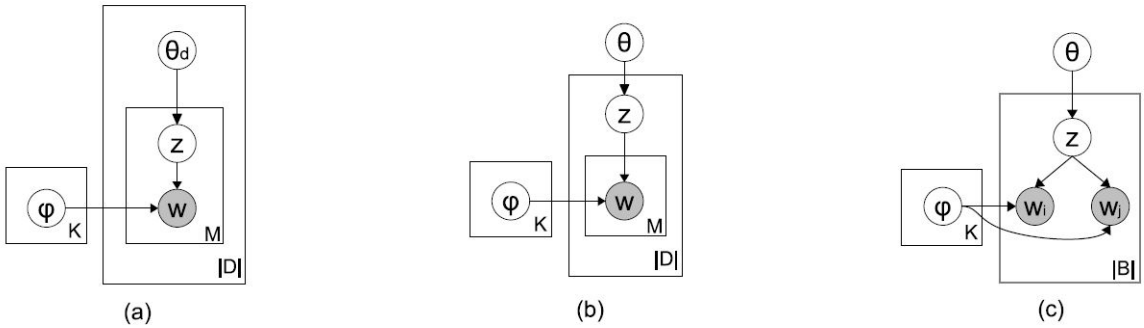

Figure 6.12: Graphical representation of (a) LDA, (b) mixture of unigrams and (c) BTM. For clarity, the fixed hyperparameters are not represented α and β.

Unlike LDA and mixture of unigrams, BTM models the biterms generation procedure in a collection, instead of documents. This comparison between the classical models and BTM is highlighted by the figure 6.12. **LDA** first generates a distribution of topics at document level in order to be able to choose between them when considering each word in the document and perform this assignment, which leads to a high dependency on the terms present in the documents and causes difficulties for short texts, where the scarcity of information can make this topic learning difficult. On the other hand, **mixture of unigrams** partially solves this problem by generating the distribution of topics at the corpus level, although it assumes that all the words in the same document deal with the same topic, which is a simplistic reduction, since even in short texts more than one topic can be dealt with. **BTM** overcomes this by dividing documents into biterms, preserving the correlation between words as well as the different topics within a document.

For its **implementation**, and given that the original work [77] only provides a series of scripts to work with, it was decided to take a library that would adapt this model for its use in Python. At first we tried to use the implementation **biterm** dated 2019 [68], due to its similarities with the one used for LDA, as even the final results are shown as a pyLDAvis notebook. However, when trying to install it, several problems occurred during the compilation of the packages, so it was decided to change the library to a more recent one (as of 2021): *bitermplus*.

6.3.1 Model training and metrics

Bitermplus is a *cythonized* version of the implementation of [77] that is also capable of calculating metrics such as perplexity or semantic coherence [63]. Thus, for each of the topics to be analyzed, it is necessary to create a specific model and train it. When creating a model using this library, at least the following parameters must be defined:

- **X**. Frequency matrix of terms and documents (*term-document matrix*).
- **vocabulary**. Vocabulary used in the given corpus, in word-list format.
- **seed**. Random seed used for the creation of the model.
- **T**. Number of topics to be extracted from the model.
- **M**. Number of most used words for the calculation of coherence.

Once the specific model has been generated, it must be trained to fit the dataset to be analyzed. To do so, it is sufficient to specify the number of **iterations** (how many times the algorithm parameters will be updated) and the biterms to be trained with, which are passed as a list generated from the **get_biterms** method, which provides it through a list containing all the documents of the vectorized corpus.

As we have seen, for the creation of the model it is necessary to specify in advance the number of topics in the whole corpus. However, in most cases, a topic extraction process is carried out when the number of topics present in the whole corpus is not known. This is why, when setting this parameter, an estimation of the ideal number of topics to be extracted must first be made. To do this, it is sufficient to generate models and train them by varying only the number of topics considered and finally choosing the one with the lowest entropy. The **entropy** can be defined as a measure of the lack of existing information [11]. Therefore, and following the principle of maximum entropy, when considering entropy as negative information, the ideal will be to minimize this metric. This is why the search for the ideal number of topics will end with the quantity that gives the lowest entropy value. Specifically, *bitermplus* performs this calculation using Renyi's entropy [37].

The measure of **perplexity** is used to quantify the degree of accuracy with which a probabilistic model predicts a sample. In NLP, it corresponds to the degree of uncertainty that a model has in predicting a text. Its effective calculation corresponds to the inverse probability of the data obtained from a test set $W = w_1 w_2 ... w_N$, normalized by the number of words. For a 2-gram model, this would be:

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Therefore, minimizing the perplexity is equivalent to maximizing the likelihood of the test set according to the linguistic model [35].

The last metric offered by *bitermplus* for model evaluation is **semantic coherence**. This measure can be interpreted as the logical relationship without contradiction between the elements (notions, propositions and topics) that make up a text.

Therefore, thanks to it, we can assess the quality of the topics detected on the basis of the co-occurrences of terms in the documents, being the degree to which a topic is "supported" by a set of texts (reference corpus). This requires a list of the most frequent terms for each topic, as these are represented to the user from such a list; $V^{(z)} = (v_1^{(t)}, ..., v_M^{(t)})$. We see that the extension of the latter has already been defined thanks to the parameter M. Thus, the calculation of the semantic coherence is done as follows:

$$C(t, V^{(z)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} log \frac{C(v_m^{(t)}, v_l^{(t)}) + 1}{C(v_l^{(t)})}$$

6.3.2 Results

For the visualization of results, *bitermplus* makes use of the library developed by Terpilowski himself: **tmplot**, a Python package for the analysis and visualization of topic modelling results. It provides an interactive reporting interface that borrows heavily from LDAvis/pyLDAvis and builds on it by offering a set of metrics for calculating topic distances and a set of algorithms for calculating topic dispersion coordinates [64]. This final result is generated as an interactive console, which makes it impossible to save it for future reference. This console consists of three plots, none of which can be omitted or it will result in problems when viewing the plot:

- **Intertopic Distance Map**. Analogous to that seen in LDA, although this time topic 0 does not represent the set of all of them, but a specific one. The distance between topics can be calculated in several ways instead of only using MDS. However, for the comparison between LDA and BTM to be consistent, the distance between topics must be fixed.

- **Most relevant terms**. Equivalent to that used in the LDA display, whose relevance can also be adjusted according to the parameter λ.

- **Most relevant documents by topic**. List of the documents with the highest presence in each topic, with a range between 0 and 100 of these.

It should be noted that interactivity has been reduced, as clicking on bubble charts or on the terms themselves no longer highlights them as it did on the LDA pages. Instead, each topic to be inspected must be chosen manually via a selector. The three graphs can be seen in figure 6.13, where the parameters relating to topic selection and adjustment of the different graphs have been omitted.

The results obtained are very similar to those obtained using LDA. Due to this fact, and together with the reduction of functionalities of the visualization notebooks, we have not carried out such a detailed study of the results obtained in this section and we have mainly corroborated the correspondence of the results with those already present in LDA.

Regarding the **results themselves**, we see that the ideal number of topics calculated on the basis of entropy is around 10 for the tweet data, while for reviews it is reduced to just 7 (which is logical as these are texts with a much lower variability of topics). While it is true that the entropy values are adequate, both the perplexity and coherence results are quite poor. This may be a problem of the data itself, which needs to be cleaned more thoroughly, or errors in the implementation of the library itself. However, due to the time constraints of the project, these results will be maintained, leaving any possible improvements as work for future iterations of the project.

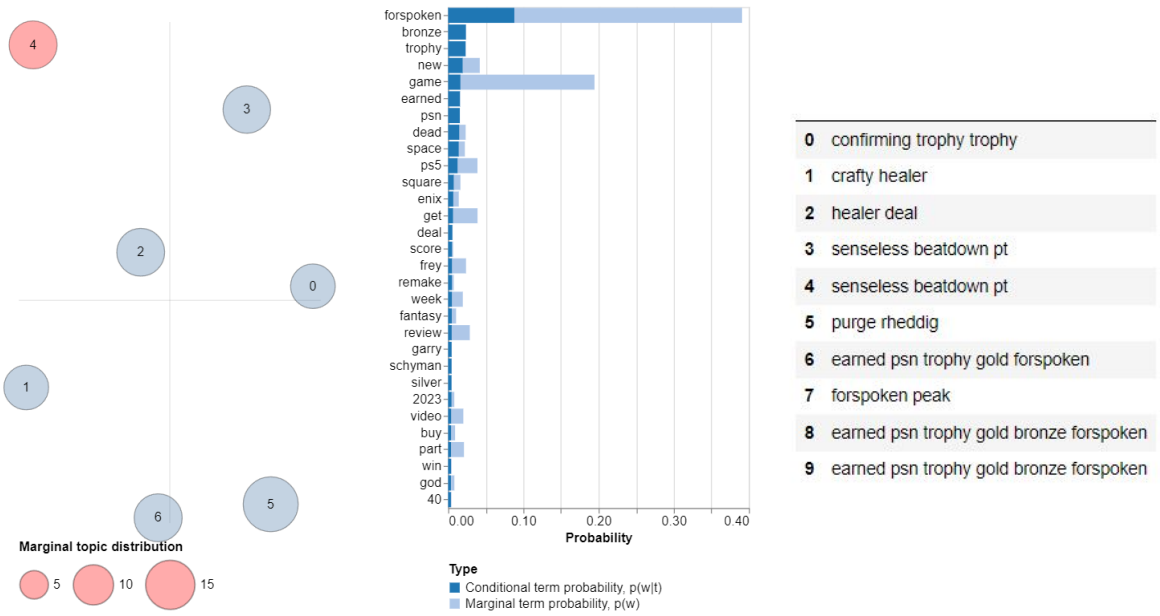

Figure 6.13: Console for visualizing the results of the model relating to tweets mentioning the videogame Forspoken, selecting topic 4 for analysis.

Despite all this, the topics obtained are still consistent and continue the line that was already defined with the LDA models, as for example in the Xbox tweets you still get topics focused on the GTA banned accounts patch, or the predominance of the Nintendo Direct in the majority of tweets mentioning Nintendo.

To mention a topic that was not discussed before, we can highlight that, in the tweets that refer to Forspoken, one topic is clearly distinguished from the rest that deals with messages from players informing via Twitter that they have obtained an achievement in the videogame. This can be seen in the figure 6.13, where in the most relevant terms we find tokens such as "-"*trophy*", "*bronze*" or "*earned*"; being in turn the most relevant documents of that topic messages that evidence the subject itself.

Chapter 7

Sentiment analysis

Sentiment analysis refers to the process of classifying the comments or opinions present in a text into categories such as "*positive*" or "*negative*", usually with an intermediate category between the two also known as "*neutral*". In the context of data science and machine learning, sentiment analysis is also called **opinion mining** or, in marketing terminology, **Voice of the Customer** (VoC). It can be a very useful tool for checking affinity towards brands, products or domains. Applied to social media, it provides an overview of public opinion on specific topics. However, it also has its limitations and should not be considered as a completely accurate reflection of reality.

Sentiment analysis is often approached as a classification problem within the field of supervised machine learning, where the assignment of the given category is based on the information contained in each text. This is a non-trivial problem with multiple challenges, many of which are related to how language itself functions, as the same word can have different connotations depending on the context in which it is used. An example of this can be seen with the term "*legend*", that could have positive connotations when referring to someone remarkable, yet in certain contexts, it might be used humorously or sarcastically: "*Wow, spilling coffee on my laptop again—what a legend.*" This task of sentiment detection becomes even more challenging when texts employ subtle nuances, humor, or sarcasm, completely altering the intended meaning of the information [17].

Nowadays, it is common to mix this approach with the use of lexicon-based methods. A **lexicon** is the set of words that make up a given lexical or linguistic modality and, by extension, the dictionaries that collect them are also referred to as such. In the context of sentiment analysis, it refers to a collection of words and phrases that are assigned sentiment scores based on their meaning and context. A good lexicon should encompass a wide range of vocabulary and expressions, as well as domain-specific and slang terms. It should also be regularly updated to reflect changes and trends in language use.

Therefore, currently employed techniques, which mix both approaches, often end up resorting to probabilistic classifiers to assign scores of positive, negative or neutral to a sentence, document or entity; as can be seen in the example 7.1, where the value *compound* refers to the general sentiment of the text known as **polarity**, whose value ranges from -1 to 1. For short texts such as those presented in this paper (tweets and reviews), the length constraints of the documents treated do not offer a distinction between sentence analysis and the document as a whole.

```
best class gift timerra fire emblem engage
{'compound': 0.7964, 'neg': 0, 'neu': 0.386, 'pos': 0.523}
```

Code 7.1: Example of assigning sentiment scores to a text

For this project, a classification of the sentiment associated with tweets and reviews will be carried out based on pre-trained models available on the web, comparing the results obtained between the different models. In the case of videogame reviews, a comparison of these results will also be carried out with other scales such as the subjectivity of the text or the usefulness of the assessment according to the criteria of other users in the community. In addition, an own model will be generated to calculate the sentiment associated with them using a probabilistic classifier.

7.1 Pre-trained models

The use of **pre-trained classification models** is a great starting point to get started in this task, as they have been trained by professionals to perform this task, giving access to a wide variety of functionalities with only a few lines of code. However, it should be noted that as these are usually developed for general purposes, they are subject to the limitations of flexibility that proprietary models do not suffer from, as it is the data scientist himself who designates the training sets to provide the desired functionality to his model. In this section we will look at three different models for general text sentiment classification and compare them with each other.

7.1.1 BERT

BERT stands for *Bidirectional Encoder Representations from Transformers*, and is a state-of-the-art pre-trained model for NLP tasks. It was developed by Google in 2018 and the first thing it does is to use an embedding like the ones presented in the section 6.2.2 to represent words as vectors. Once this is done, they are passed to a deep neural network with multiple layers of transformers to encode the meaning and context of words and phrases from both directions. BERT can be refined for specific tasks, such as sentiment analysis, by adding a classification layer on top of the pre-trained model and training it with labelled data [31].

There are variants of this model such as **RoBERTa** (Robustly optimised BERT approach), which uses BERT as a base by modifying some of its hyperparameters, removing the pre-training target from the next sentence and training with **mini-batches**[1] and learning rates much higher.

For this project, a RoBERTa model trained on some 58 million tweets and refined for sentiment analysis of tweets written in English has been used, following the TweetEval framework [9]. Broadly speaking, it works by analyzing each text individually, giving a score between -4 and 4 for each of the existing categories (positive, neutral or negative), indicating the intensity of each of these sentiments in the text. These scores are then normalized and converted into a final percentage for each of the categories.

```
making progress finally let play fire emblem engage
[-3.4069092  1.1576403  2.1524656]
[0.00280364 0.26920322 0.7279932 ]
```

Code 7.2: Scores and percentages (negative, neutral and positive; respectively) assigned by RoBERTa to a tweet.

The main **drawback** of this model lies in its **speed**, as the assignment of a score makes use of tensors to analyze the text, slowing down the final calculation. In addition, and contrary to what happens with the other models that will be seen, it does not include a functionality to calculate an aggregate score that indicates the general feeling conveyed by the text. Therefore, with this in mind, two metrics have been considered in order to evaluate this final sentiment:

- **maxROBERTA**. The category with the highest percentage is chosen as the predominant sentiment; labelling 0, 1 or 2. However, this screening is somewhat crude and does not take into account the greater weight of the positive (2) and negative (0) categories compared to the neutral (1), so many tweets are classified in the latter.

- **compoundROBERTA**. Taking the scores given by the model, we can normalize these to calculate a function that elucidates the final sentiment based on these values:

$$f(pos, neu, neg) = \begin{cases} 0, & \text{if neu} > |\text{pos - neg}| \\ \frac{pos+neu-neg}{\sqrt{(pos+neu-neg)^2}}, & \text{in any other case.} \end{cases}$$

  where *pos*, *neu* and *neg* are the unnormalized scores of the positive, neutral and negative sentiment present in the text. Thus, in case neutral were the predominant sentiment (since its score is never negative), we give it a value of 0. Otherwise, we normalize the scores and see if the sentiment obtained is positive (1) or negative (-1). The idea of this function is to emulate the polarity function provided by the other pre-trained models.

[1] When training a machine learning model, the **batch size** is a hyperparameter that defines the number of samples to work with before updating the internal parameters of the model. A training set can be divided into one or more batches [16].

7.1.2 VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically tailored to sentiment expressed in social media [33]. To use it, simply use the NLTK package and import the **SentimentIntensity-Analyzer** model, making sure that the downloaded VADER lexicon is available on your computer. Once the model has been instantiated, simply apply it to the individual texts.

Its main **advantage** is that it is very easy to use and calculate, as well as quite fast. Moreover, apart from the specific percentage of sentiment expressed in each tweet, it also gives a composite score that allows to classify it as positive, neutral or negative with greater precision. This value is called **polarity**, which is a normalized score that ranges from -1 to 1 and indicates the underlying positive or negative sentiment in each text.

Since VADER has been specifically trained to detect sentiment associated with social media texts, it will be used to analyze both tweets and reviews extracted. In this case, texts that give a score of 0 will be considered neutral, and the other cases will be considered positive or negative (depending on whether the result is positive or negative, respectively).

7.1.3 TextBlob

TextBlob is a Python library for textual data processing. It provides a simple API to dive into common natural language processing tasks such as sentiment analysis, detecting the underlying subjectivity of a text, or translating [40]. When calculating sentiment, TextBlob takes the average of the whole text, so that only the common meaning of a word in the whole text is taken into account. This task is performed using the WordNet database and from a NLTK model trained on a corpus of movie reviews. For this reason, it will be used only to calculate the sentiment associated with videogame reviews obtained from Metacritic.

One of the advantages of TextBlob is the possibility to make use of additional functionalities apart from sentiment detection itself, such as the calculation of the degree of subjectivity present in a text. **Subjectivity** is used to determine whether the analyzed text expresses an opinion or not. Mathematically, it is represented as a value between 0 and 1. The higher the value, the closer to an opinion the text is considered to be. Although all the texts considered are videogame reviews, and therefore have a certain degree of intrinsic subjectivity associated with them, we will want to see if the reviews that make use of more personal assessments have any influence on the polarity of the final text (as seen in figure 7.11).

7.2 Specific models

Sentiment detection of a text is performed after having previously trained sentiment classifiers. So far, we have only used pre-trained classifiers with general or specific texts. The main advantage of training your own model is that you can decide the training sets to be as similar as possible to the real data you will end up using, giving better results. In this section, we will also train our own classifier to identify whether a review is positive, negative or mixed. We perform this task only on videogame reviews, as all the data is pre-labelled in one of the three categories, so training the models can be easily done.

7.2.1 Naive-Bayes

To perform the classification task, we will make use of a **multinomial Naive-Bayes Classifier** (**NBC**), which is a Bayes classifier whose basic idea is to simplify hypotheses about how features interact with each other. The underlying idea behind this is that of a **bag of words**, which consists of considering the text of a document as an unordered set of words in which the position of the words is ignored, keeping only the frequency of occurrence of the terms. Figure 7.1 shows an example of this process from an English review of a film [35].

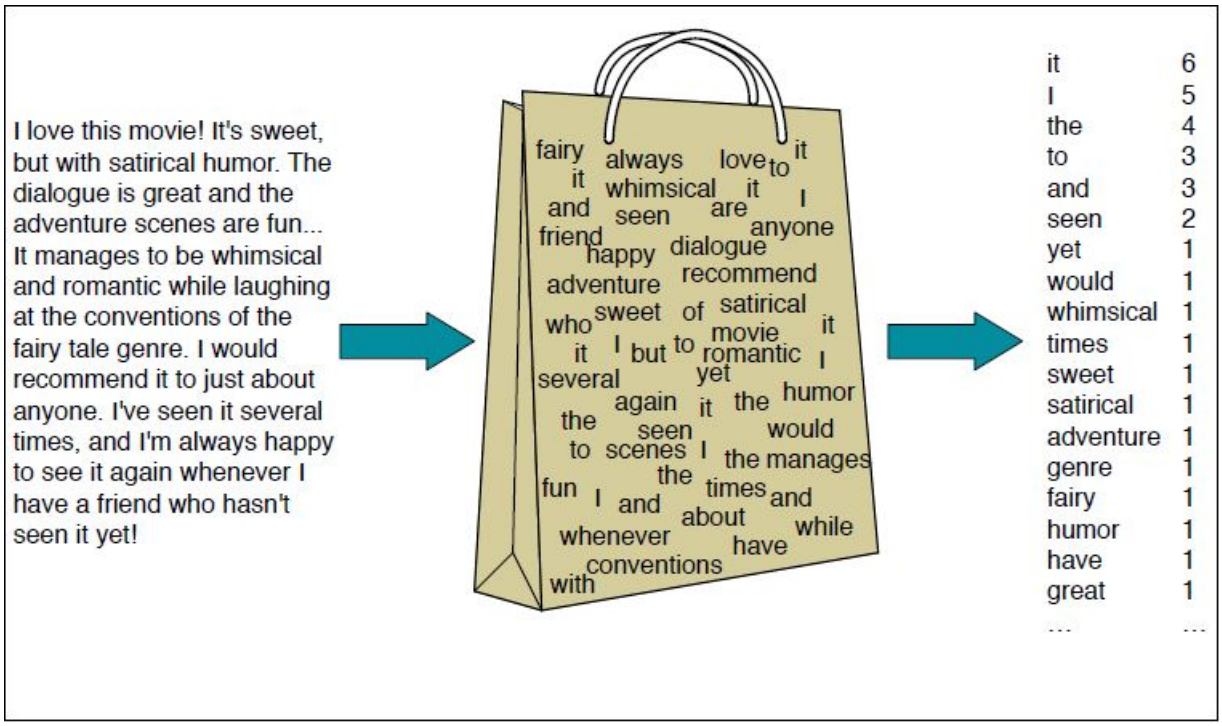

Figure 7.1: Intuitive idea behind the Bayes classifier and bag of words

Naive-Bayes is a probabilistic classifier, which means that for a document d, of all classes $c \in C$, the classifier returns that class d which has the maximum posterior probability given the probability of the document. For this calculation, we use the **Bayesian inference**, whose idea is like the one used in the n-grams: simplifying the calculations by applying the conditional probability rule (discarding the probability $P(d)$ as it is a constant) and assuming that the different features that represent a document $(f_1, f_2, ..., f_n)$ are independent of each other (**Naive-Bayes hypothesis**).

$$\tilde{c} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} \frac{P(d|c)P(c)}{P(d)} = \arg\max_{c \in C} P(d|c)P(c) =$$

$$= \arg\max_{c \in C} P(f_1, f_2, ..., f_n|c)P(c) = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i|c)$$

Specifically, the features f_i are all the possible positions w_i that a word could occupy throughout the text. Moreover, as with the n-grams, these calculations are performed in logarithmic format to avoid problems of **underflow**, so that the final equation results:

$$\tilde{c} = \arg\max_{c \in C} \log P(c) + \sum_{i \in \text{positions}} \log P(w_i|c)$$

Therefore, the final class prediction is made as a linear function from a set of input data. Classifiers that use a linear combination of the inputs to make a classification decision are called **linear classifiers**.

7.2.2 Model training

Once the model has been defined, it needs to be trained on the data we have so that it can perform the desired classification tasks. However, not all of the data is used to train the model. Ideally, about 60 or 80% should be used to train the model (**training dataset**), with the remainder available to evaluate the quality of the model obtained (**test dataset**). In addition to this, it is advisable to **stratify** (divide the data into groups or strata according to their characteristics) the data so that their final distribution is as similar as possible to that of the future samples to be predicted. Once these parameters have been defined, it is also necessary to specify others, such as the batch-size or the number of epochs[2].

However, the question arises as to how these data are used to train the model. Returning to the expression seen above, it is most common to estimate the probabilities $P(c)$ and $P(f_i|c)$ by maximum likelihood estimation: using the relative frequencies of the training data.

Thus, if N_c is the number of documents of class c in the training dataset and N_d is the total number of documents, we can estimate $\tilde{P}(c) = \frac{N_c}{N_d}$.

[2]The number of **epochs** is a hyperparameter that defines the number of times the learning algorithm will work through the entire training dataset, updating the internal parameters of the model. An epoch is composed of one or more **batches** [16].

Feature estimation is performed in an analogous way, being the frequency of occurrence in the bag of words for documents of class c. In addition, to avoid null probabilities, the same trick as used in N-grams is repeated, adding 1 as a default value for each count[3]:

$$\tilde{P}(w_i|c) = \frac{C(w_i, c) + 1}{\sum_{w \in V}(C(w, c) + 1)} = \frac{C(w_i, c) + 1}{\sum_{w \in V} C(w, c) + |V|}$$

where $C(w_i, c)$ is the fraction of times that the word w_i appears among all words in all documents of class c and V is the vocabulary of the corpus.

7.2.3 Model evaluation

After having trained the model, it is necessary to evaluate its performance in order to check its effectiveness. To do this, the test dataset is used and various metrics are extracted from the results obtained, which are calculated according to whether or not the classification task has been performed correctly:

- **True Positive** (**TP**): Correctly predicted values (correct prediction).
- **False Positive** (**FP**): Incorrectly predicted values (incorrect prediction).
- **True Negative** (**TN**): Correctly rejected values (correct prediction).
- **False Negative** (**FN**): Incorrectly rejected values (incorrect prediction).

Although these values are particularly intuitive in the binary case, for the n-class situation it is sufficient to perform this analysis on a class-by-class basis. These results can be summarized using the technique known as **confusion matrix**, which provides information about the hits and misses of the classification model, as well as the type of errors it makes. Figure 7.2 shows the confusion matrix generated after evaluating the results obtained from the model capable of classifying user reviews. In it, we see that none of them have been predicted as "*Mixed*", due to the fact that there is also not a large amount of data to train and, as we will see in section 7.3.3, user reviews tend to be highly polarized.

Returning to the model evaluation, based on these values, various metrics can be obtained that provide information on the validity of the model obtained:

- **Accuracy**. Frequency with which the evaluated method makes the correct prediction. It is calculated as the sum of the true predictions divided by the total number of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Recall**. Fraction of positive cases that were predicted to be positive.

$$\text{Recall} = \frac{TP}{TP + FN}$$

[3]This is known as **Laplace smoothing**.

- **Precision**. It represents the accuracy of the method. It is calculated as the proportion of cases that were predicted to be positive and were actually positive, divided by the total number of cases that were predicted to be positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **F-score**. A metric that combines recall and precision to determine the accuracy of the test. For n-class problems, this score is usually found for each class. It is calculated as:

$$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- **Support**. Number of observations that are predicted in a given class.

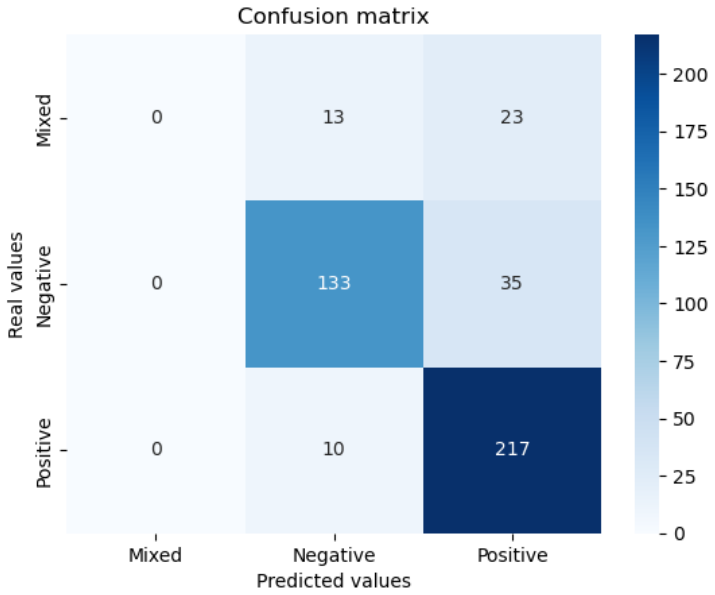

Figure 7.2: Confusion matrix of the model trained with user reviews to predict the category of a review

In addition to all these metrics, it is also common to make use of **cross-validation**, a technique used to evaluate the results of a statistical analysis and ensure that they are independent of the partition between training and test data. The most common way to implement this is the method known as **K-folds**, in which the input data is split into K parts, one of which is reserved for testing and the other $K - 1$ for training. This process is repeated K times and the evaluation metrics are averaged. This helps determine the generalization of a model to new datasets.

With regard to the **results obtained in the notebooks**, we can see that the main problem has been the scarcity of data for training, as well as its bias. We have already mentioned the scarcity of user reviews with intermediate ratings in the figure 7.2, and the reviews in the trade press suffer from the same problem for the negative ratings. However, in both cases, final models have been obtained with an accuracy of between 70% and 80%, so both classifiers can be considered to perform acceptably.

7.3 Results

Having seen the theoretical framework on which the sentiment analysis work is based, it is now possible to scrutinize the results obtained. Due to the large number of graphs and statistics generated, only some of them will be commented on, and the rest can be consulted in the notebooks included as attached content. Furthermore, in this section we will only discuss the results obtained with the **pre-trained models**, as the classifiers created and their accuracy have already been discussed in the section 7.2.3.

The analysis is divided according to the data considered, distinguishing tweets from reviews. In the latter, a distinction is also made as to whether they are reviews written by trade press or by the community itself, since, as could be seen in studies such as Pellarolo's (figures 3.3 and 3.4), there is a palpable difference between the evaluations given by both groups when it comes to giving scores.

7.3.1 Tweets

In the case of tweets, we can only compare the results obtained by the three metrics defined with the RoBERTa and VADER models. Although RoBERTa has been specifically trained to detect sentiment on tweets, not having a defined metric to perform classification tasks penalizes its performance compared to what the VADER model offers.

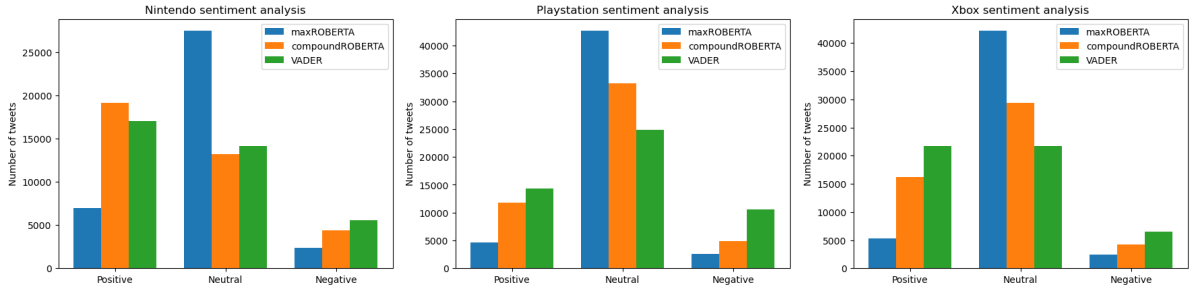

Figure 7.3: Sentiment detected according to the metrics defined for RoBERTa and VADER in tweets mentioning Nintendo, Playstation and Xbox

As can be seen in figures 7.3 and 7.4, the maxROBERTA metric tends to classify almost all texts as neutral. The score created in this study, compoundROBERTA, offers better results, which are also similar to those obtained using VADER. In any case, the three companies have an equivalent number of both positive and neutral tweets, with significantly fewer negative tweets, so it can be deduced that all three brands have a positive perception by the community. In the case of their respective releases, both Hi-Fi Rush and Fire Emblem Engage receive mostly positive feedback, while Forspoken seems to have had a more tempered reception from users, with about half as many positive tweets mentioning it as its competitors.

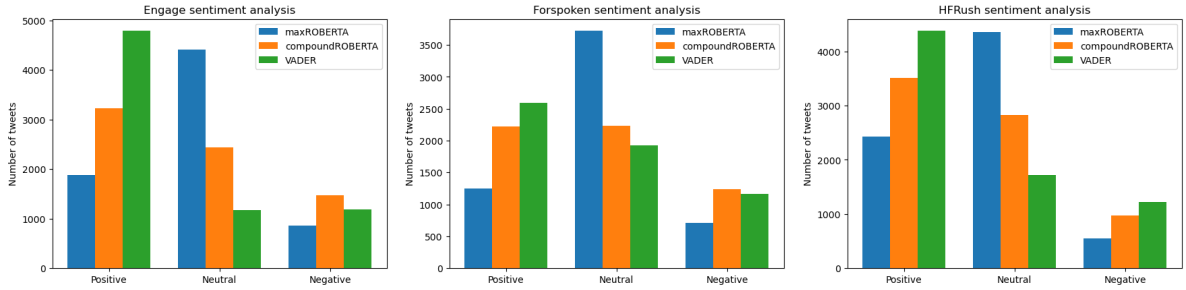

Figure 7.4: Sentiment detected according to the metrics defined for RoBERTa and VADER in tweets mentioning Fire Emblem Engage, Forspoken and Hi-Fi Rush.

In addition, the effectiveness of the classification models can be tested by obtaining a random sample of the tweets considered. For this purpose, at least one tweet has been selected from each category designated by the maxROBERTA metric, as it seems to detect only the most polarized tweets. If we look at the figure 7.5, practically all tweets have been properly categorized considering their text. However, some discrepancies can be observed, as in the case of the tweet about Hi-Fi Rush, which both ROBERTA metrics have detected as negative, but VADER categorizes it as positive. This is explained by the aggressive language of the tweet, which does not make the predominant sentiment of the text very clear, although once analyzed carefully it can be seen that it has positive connotations, praising the product for its quality.

	text	ROBERTAmaxSentiment	ROBERTAcompSentiment	VADERsentiment	tweet
15440	fuck nintendo the greedy'assholes	0	-1.0	-0.7003	Nintendo
37540	Check out my broadcast from my PlayStation 4! #PS4live (The Last of Us™ Remastered) live at https://t.co/CiBZEurt5M	1	0.0	0.0000	Playstation
21840	Will be raiding an non affiliate in a few hrs come say hi come watch us win \n#twitchstreamer #StreamersConnected #streamer #xbox #fornite #nonaffiliate #SupportSmallStreams #support #hype #love \nhttps://t.co/pXk0zk4UyH	2	1.0	0.5859	Xbox
4311	While I really appreciate the weapon triangle returning in Engage and it also changing how you interact especially early game. It's so strange to me when I see relatively new FE people talk about how they were pissed off 3H didn't have	2	1.0	-0.5186	Engage
3766	Finally have a little time to dive into #Forspoken\nThis is fun! Not sure what some people were complaining about. #Frey is not amused. https://t.co/WhRKXRoxaP	2	1.0	0.7458	Forspoken
5146	hi-fi rush is a goddamn banger\nno early access, no gb bugs, got released the day it got announced, music slaps, free on the gamepass (or 30 euros), fun gameplay, characters and story (fuck corporations)	0	-1.0	0.1531	HFRush

Figure 7.5: Sample of some tweets and the ranking given by each pre-trained model

7.3.2 Trade press reviews

The main problem with the press reviews for the titles collected is the paucity of data for proper comparisons. Moreover, almost all of them do not usually give a score that can be categorized as negative, which was also a problem when training the classifier in the section 7.2.3. For this reason, the analysis of subjectivity plots or box plots of how the reviews are distributed with respect to polarity will be discussed in the next section, where more data already exist to give such a comparison.

However, another comparison can be made. The main advantage of the reviews is that they include a **numerical score** for each product. If this value is normalized, it can be used to compare the score given with the actual text of the reviews, checking whether this score corresponds to what the text conveys. To do this, and given that the press evaluations give a numerical score between 0 and 100, the following **scaling** must be carried out in order to compare it with the polarity of the texts (value between -1 and 1):

$$f(x) = (\frac{2x}{100} - 1) \in [-1, 1] \text{ para } x \in [0, 100]$$

Figure 7.6 shows these graphs for each company's products on their own platforms. As with RoBERTa, TextBlob seems to have some tendency to opt for intermediate values, without overly polarizing the final score. At the other extreme is VADER which, when compared to the standardized scores given, seems to exaggerate the sentiments present in the text, although its approximation is more accurate than TextBlob's. In any case, Forspoken seems to be the only product of the three that has obtained mediocre or not so positive ratings, unlike the other two videogames, which have obtained very favourable reviews.

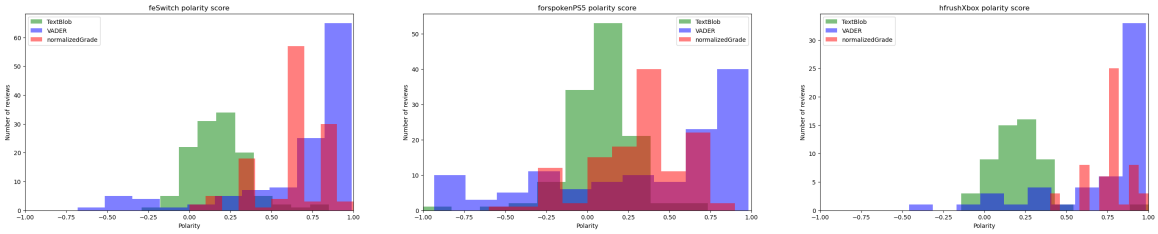

Figure 7.6: Comparison between the polarity of the text and the rating given by the trade press to Fire Emblem Engage (Switch), Forspoken (PS5) and Hi-Fi Rush (Xbox Series X)

VADER's inaccurate behaviour is due to the fact that it has been trained to recognize the sentiment of social media texts. Press reviews, which have a more formal tone than social media texts, are more difficult for it to locate and that is why it makes such errors. For user reviews, however, it does perform well, as shown in the figure 7.8.

Finally, considering the subjectivity ratings provided by TextBlob (figure 7.7), we see that intermediate values are obtained in practically all cases. However, the scarcity of reviews does not allow conclusions to be drawn from the data currently available, so this assessment will be postponed until this same scrutiny is carried out with the opinions of users, which are more numerous and will allow us to obtain clearer results both in this graph and when comparing the polarity and subjectivity values detected.

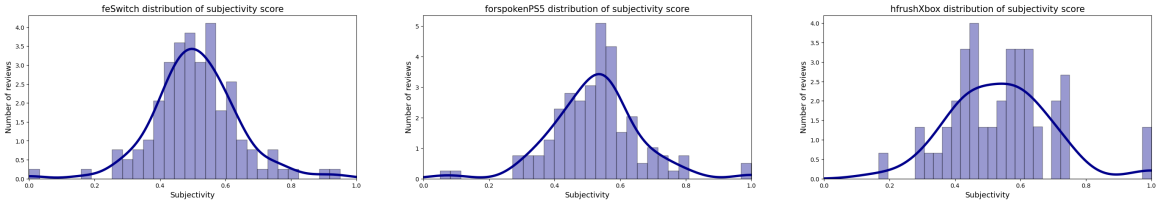

Figure 7.7: Distribution of subjectivity found in trade press reviews of Fire Emblem Engage (Switch), Forspoken (PS5) and Hi-Fi Rush (Xbox Series X)

7.3.3 User reviews

The polarity comparison in the case of users is carried out in an analogous way, although this time the scaling function must be adapted to the users' grades (instead of being over 100 it is over 10):

$$f(x) = (\frac{2x}{10} - 1) \in [-1, 1] \text{ for } x \in [0, 10]$$

The figure 7.8 shows even more markedly than in the case of the press reviews how poorly the Forspoken videogame has been received, while Hi-Fi Rush has been uniformly recognized as a great product.

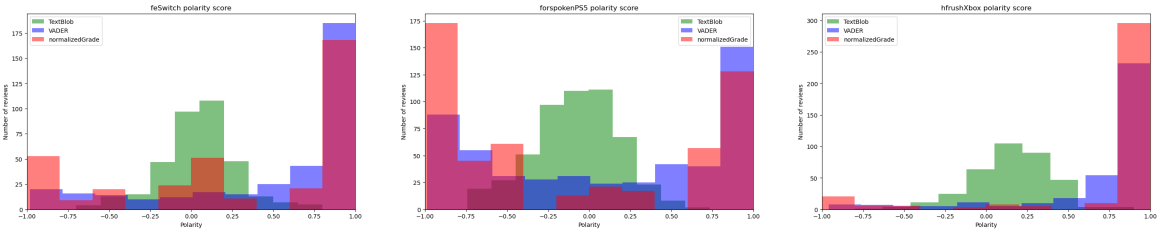

Figure 7.8: Comparison of text polarity and user ratings for Fire Emblem Engage (Switch), Forspoken (PS5) and Hi-Fi Rush (Xbox Series X)

These results are even more evident in figure 7.9, as the negative Forspoken reviews have lower polarity values, while the positive Hi-Fi Rush reviews show higher polarity scores.

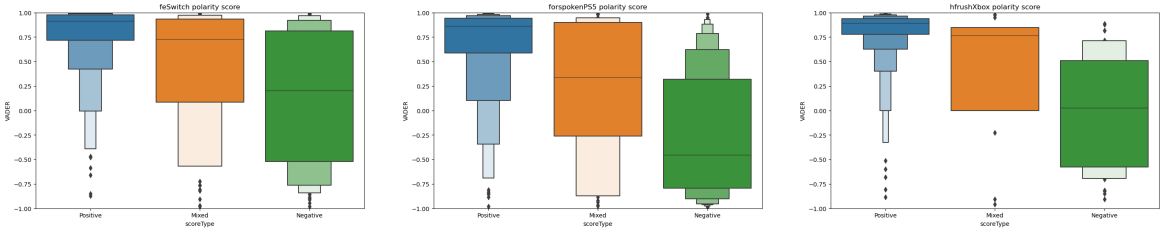

Figure 7.9: Box plots indicating the polarity of the texts according to the type of review

Returning to the analysis of subjectivity from the previous section, we see that on this occasion we do have sufficient data to be able to construct appropriate subjectivity distribution histograms (figure 7.10). Once again, the overall values are fairly focused, although for Hi-Fi Rush we have detected a little more subjectivity in the ratings than usual.

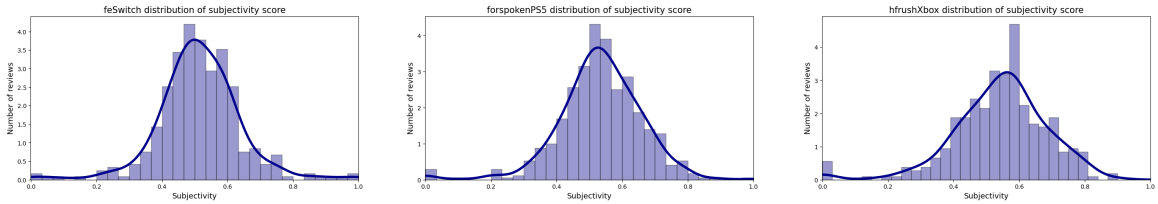

Figure 7.10: Distribution of subjectivity in user reviews of Fire Emblem Engage (Switch), Forspoken (PS5) and Hi-Fi Rush (Xbox Series X)

With respect to the influence of a subjective assessment of text polarity, we see in figure 7.11 that the vast majority of criticisms are located in the central area of the graph, with intermediate values for both polarity and subjectivity. This is due to the fact that the polarity metric used is the one provided by TextBlob, which usually gives values that are not excessively polarized (figure 7.8), so it might be convenient to repeat this analysis with the polarity values given by VADER. However, it has been preferred to keep this analysis so that both metrics were those provided by the same model.

Nevertheless, it can be observed that Forspoken has a higher number of subjective ratings with lower polarity, while Hi-Fi Rush has obtained more polarized reviews while maintaining above-average levels of subjectivity. Fire Emblem Engage, meanwhile, retains most of its reviews in the middle range.

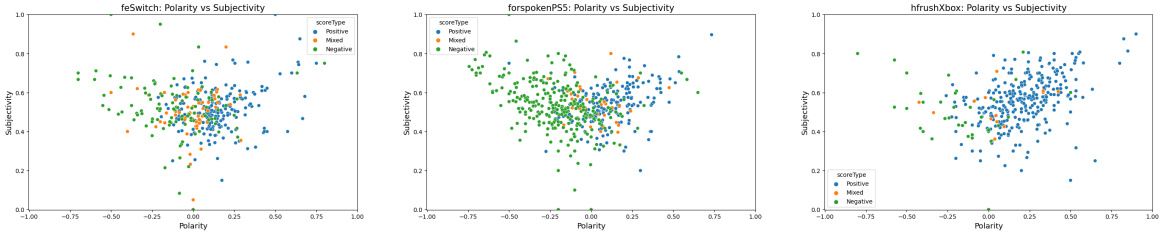

Figure 7.11: Comparison between polarity and subjectivity values detected by TextBlob in user reviews of Fire Emblem Engage (Switch), Forspoken (PS5) and Hi-Fi Rush (Xbox Series X)

In addition, user ratings include an additional metric of usefulness or **helpfulness**, which indicates whether a review has been useful to the rest of the community. For example, for the case of the videogame Fire Emblem Engage we see that those reviews that have been most useful to the community are categorized as intermediate or negative, but maintain polarity-centered values (Figure 7.12). This is due to the fact that this is a videogame saga with a large community of followers, so a part of the positive reviews are comments from fans praising the game, which is not particularly relevant information for a neutral player who wants to learn about the product to decide whether to purchase it or not.

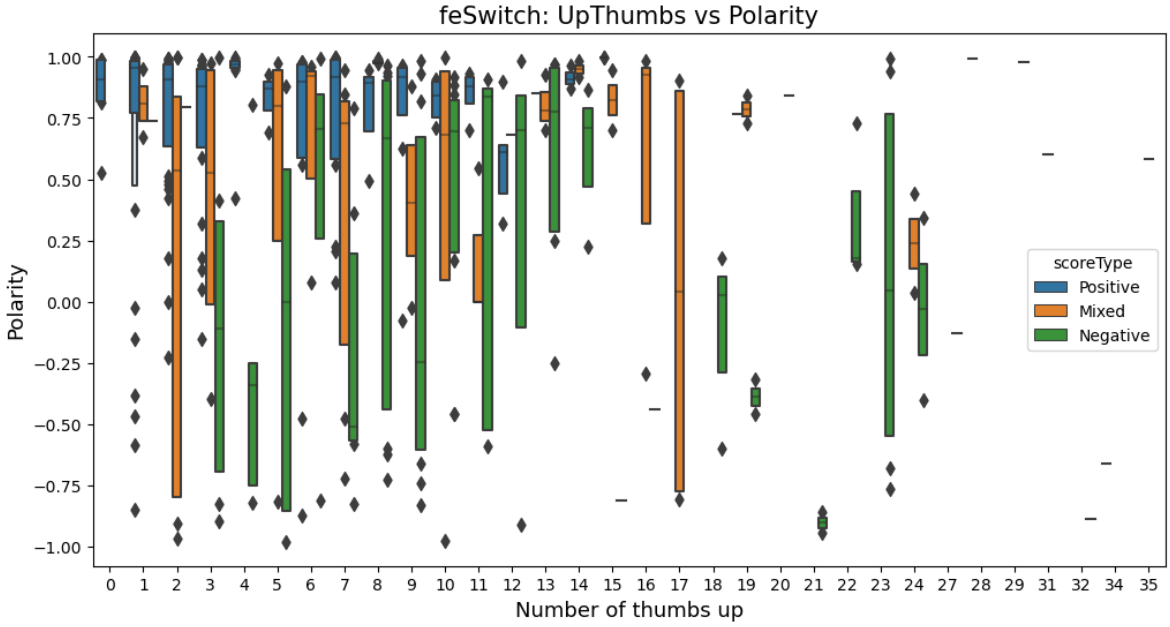

Figure 7.12: Comparison between the usefulness of the Fire Emblem Engage videogame reviews and the polarity detected by VADER, as well as the category in which they fall into

Chapter 8

Conclusions and future work

This section will draw conclusions both from the products and brands analyzed throughout this work, as well as the learning that the student has gained from the development of the project. It also includes possible improvements and alternative lines of work should future iterations of the final study be carried out.

8.1 Conclusions

8.1.1 Conclusions of the study

Once the project has been completed, the only thing left to check is whether the objectives set at the beginning of the project have been achieved. Let us analyze point by point whether this has been the case:

- **OBJ-1:** A total of seven notebooks have been generated, as well as a final report, covering all the steps and final products that would make up the development of an NLP data science project.
- **OBJ-2:** Although the relevance of the issues identified was subject to current events, the process of extracting issues has also been carried out satisfactorily (chapter 6).
- **OBJ-3:** After consulting the data collected on social networks and examining the different reviews, we have seen that Fire Emblem Engage and Hi-Fi Rush have had a mostly positive reception, unlike Forspoken. Forspoken hasn't been met with widespread rejection either, but it has had a more lukewarm reception from the press and community, with its weaknesses being highlighted over its virtues.

Only **OBJ-4** would be missing, which can now be assessed as almost six months have passed since the launch of the three titles and information on the general reception of each of these products is now available. However, it should be noted that, due to company policies, only a small part of this data is made available to the general public, so the main source of information is both statements from the studios themselves and pages collecting sales data on platforms such as Steam (SteamDB).

In the case of **Fire Emblem Engage**, which was released only for Nintendo Switch, it reached 1.61 million copies sold worldwide in March 2023 [30]. These results can be considered a success considering that the previous instalment of the series, Three Houses, had become the best-selling game in the franchise after reaching 3.82 million copies sold worldwide in December 2021 (two and a half years after its release). This corroborates the good data that has been obtained about the game over the course of the study.

Forspoken, on the other hand, has not been as fortunate. The earnings report published in February 2023 by Square Enix, parent company of Luminous Productions, responds to the bad reviews and negative feedback the game has received, mentioning that the game's sales have turned out to be "lacklustre". [76]. Due to this, Luminous Productions has been reinstated within Square Enix in May this year, while continuing to work on patches to improve bugs in the game, as well as the latest Forspoken DLC that was released in June [66]. However, the latter still seems to suffer from the same problems that plagued the base game.

This failure becomes even more apparent when looking at its PC sales recorded on the Steam platform, where it hasn't even made it into the top 10 best-selling games. Particularly glaring is the comparison with **Hi-Fi Rush**, a game that enjoyed virtually no associated promotional events or marketing campaign, but still managed to sneak in as the eighth best-selling game on Steam the week of its release [38]. This fact is especially serious when you consider that Forspoken cost twice as much to develop as Hi-Fi Rush [8], the latter being available to Xbox Game Pass users at no extra cost, which also allows the title to be played on PC.

Given these data, it is worth asking whether the conventional development model of big videogame blockbusters may be starting to become unproductive, due to the high risk and development costs associated with these projects. As reflected in the sales of both games, as well as the impressions gathered throughout this study, a paradigm shift to more content-driven development that prioritizes quality over length might be more profitable, resulting in more affordable products for the end user (as Forspoken costed 70$ at launch while Hi-Fi Rush was available for less than half the price [65]). The same can be applied to conventional distribution models as, in cases such as Hi-Fi Rush, we see that the statements made by Microsoft CEO Phil Spencer about the benefit in the final sales of titles released on Game Pass have turned out to be true [73].

8.1.2 Personal conclusions

Analyzing this work as just another subject in the degree, I believe that this project has allowed me to grow enormously as a professional. Although throughout the degree course there are different assignments to help us acquire competences and skills in our field, I think that none of them has been as useful to me as this project. In my opinion, one of the main reasons why this work has been so useful to me is the freedom I have enjoyed throughout its development, both by the company and the university, both institutions being embodied in the figures of my tutors, who have always resolved the doubts that arose and encouraged me to continue along the different lines of development that occurred to me for the study. This is evident in subjects such as the topics chosen to study, as analyzing themes that I am passionate about (such as the videogames industry or football) has meant that many days I have been able to work with more enthusiasm than I would have done if I had chosen any other subject. In addition, having been, for practical purposes, the person ultimately responsible for the development, I think it has also helped me a lot, as I have had to manage and organize all the tasks from the beginning to be able to adapt to the deadlines set, which I am sure will be key in my professional future.

Although I had not taken the *Sistemas Inteligentes* course, which is the closest to this work as it presents the basic concepts of Machine Learning, it is also true that I cannot be considered a complete neophyte in this field, as I have developed my work experience in this field. In fact, this phase has been the one that has allowed me to acquire the fundamental notions on which I have been able to investigate further throughout this work. In particular, I found the work carried out in data science very interesting and I will certainly try to continue my training in this field, especially using NLP techniques, which I have seen have a strong mathematical basis that I think I understand better thanks to the training I have acquired in the degree in Mathematics.

8.2 Future work

Ultimately, it leaves some points on which the work could be improved in future iterations of the project, as well as alternative development ideas taking advantage of all the data collected during the development of the project:

- **Conduct a similar study using the data collected by Twitter for the clubs in the Premier League**. As an alternative theme, this problem could be approached to take advantage of the large amount of data from Twitter relating to these teams and their signings. Such an alternative study could encompass other branches of data science, such as the performance analysis of players signed before and after they changed teams.

- **Obtain social data from other sources**. Such as Instagram, Reddit or other social media. This would allow for a wider variety of information to be obtained and better results to be obtained.

- **Implementing a non-relational database for the management of the information collected**. Due to the small scope of the project and the fact that all the development has been done locally, we have always worked with the data itself, always having a couple of backup copies of the data. However, in a more ambitious project, a non-relational database could be used to store the different collections of information available.

- **Polishing data cleaning and screening**. Although this task has already been done several times, there is always room for improvement. For example, the terms used for Twitter searches could be removed to avoid redundant results.

- **Adjusting the number of LDA topics to the ideal number of them**, in the same way as was done with the BTM model. In fact, such values could be used to adjust the number of topics to be searched.

- **Get results with better visualization for the BTM model**. One of the main weaknesses of the study is the difficulty to consult the results obtained by the biterm model, so it would be convenient to look for another library that implements it and migrate the code to these functionalities.

- **Sentiment analysis using capital letters and emoticons**. Some of the main communication tools on social media are the use of capital letters to emphasize words or emoticons to convey sentiment. However, both were screened out during the clean-up process, so their impact could not be measured.

- **Collecting more data to train review sentiment classifiers**. It would be enough to collect reviews of various releases, as Pellarolo does in his work [52].

- **Compare the subjectivity values perceived with TextBlob with the polarity values detected by VADER** as the latter, at least in the case of user reviews, were a fairly accurate reflection of the scores given.

Part III

Appendixes

Appendix A

Installation Manual

This section shows the necessary steps to emulate the development environment in which the project has been carried out, in order to be able to execute the final development of the project. The process has been carried out on a computer with Windows 10 as the operating system, although for other systems the steps to follow are practically equivalent except for slight modifications depending on the operating system used.

1. **Install the Anaconda development platform**. To do this, simply download and run the installer provided in [6], keeping the default options it offers. **Anaconda** is a free and open source distribution of the Python and R programming languages, used for data science and Machine Learning. It provides a complete, ready-to-use environment that includes a large number of packages, libraries and tools that are commonly used in the field of data science, as well as an intuitive interface that makes it easy to use.

2. **Download all the packages and libraries required by the different notebooks**. Once the previous step has been completed, you only have to look for *Anaconda Prompt* among the available programs and run it as **administrator**, since during the process some of the packages that Anaconda includes by default will be uninstalled. Opening it will bring up a command console from which the various modules downloaded to the system can be modified. The downloading of new packages is done by the command

   pip install <package_name>

   The list A.1 shows all the commands needed to perform this download, so it is enough to copy and execute them one by one. The main problem lies in installing the **bitermplus** module, which requires a version of Microsoft Visual Studio C++ equal to or higher than 14.0, for which you just need to follow the steps given at [22].

```
$ pip install tweepy
$ pip install langdetect
$ pip install wordcloud
$ pip install pyLDAvis --user
$ pip install bitermplus
$ pip install tmplot
$ pip install tomotopy
$ pip install textblob
```

Code A.1: Commands to install all packages required by the project

3. **Open the interface that gives access to the development notebooks**. Access to the development notebooks can be done in a simple way by searching for the programme *Jupyter Notebook*. When running it, a browser must be selected on which to access the display of the documents (during development, Google Chrome was used as the default option). Once this has been chosen, simply go to the location where the project is stored and run the notebooks from the start (the first one does not work, but as the core image is saved, you can see the results of when the notebook was successfully executed), as these include hyperlinks to move on to the next one in an orderly way. In fact, because the results of the last execution are kept, the notebooks can be consulted as they are, as some of these processes require quite a long execution time. In case you want to test it on the machine itself, the execution of each of the cells can be done thanks to the *Run* button (figure A.1).

Figure A.1: Header of a file executed by Jupyter Notebook

Appendix B

Content attached

This section details the contents of the attachments submitted together with the project report.

- **Development notebooks**. Jupyter Notebooks in which the various phases of the study have been implemented. At the request of the end client, they are written in English. They are a set of 7 numbered notebooks representing the various phases of the project:

    1. *RestAPI*. Extraction of data from Twitter. Due to changes in the company's access policies in recent months, **can no longer be run** as there is no official account for downloading data from Twitter. Moreover, if it could be run, it would not yield the same data either, as only tweets up to a week old can be retrieved. This is explained in the section 4.1.
    2. *WebScraping*. Extraction of data from Metacritic. Covers the section 4.2 of this document.
    3. *DataSctructure*. Notebook to visualise and better understand the structure of the data we are working with. It corresponds to the section 4.3 of the memory.
    4. *DataCleaning*. Chapter 5 of the memory. The previously obtained data are cleaned and converted to the format to be used in subsequent notebooks.
    5. *TopicModelling*. Exploratory data analysis and topic modelling from LDA (the first two sections of the chapter 6).
    6. *Biterm*. It corresponds to the section 6.3 of the chapter on topic modelling. Topic identification using the BTM model and the bitermplus implementation. However, when trying to run it on a different computer than the one used during the development, the **model display gives errors** because of the tmplot module, due to the fact that the bitermplus package uses an older version of the tmplot module. This is an extra reason to migrate the development to another BTM implementation in future iterations of the project.
    7. *SentimentAnalysis*. Detection of the sentiment associated with the different texts available, corresponding to chapter 7 of this document.

- **data**. Set of reviews obtained from the Metacritic website in the second notebook. It follows the data structure discussed in 4.3.2.
- **Furbo**. Information extracted from Twitter thanks to the first notebook, in relation to the teams belonging to England's Big Six, as well as their most recent signings.
- **Modelos**. Models obtained in notebooks 5 and 6 for topic extraction (LDA and BTM), as well as the classifiers trained during the last notebook to identify the type of review according to the text of the review. In the case of LDA, the interactive html visualization files are also included, which can be accessed from the very notebook in which they were generated.
- **Videojocs**. Information extracted from Twitter thanks to the first notebook, in relation to the three main companies in the videogame industry, as well as their releases at the end of January 2023. Also included is the sifted version of both reviews and tweets, generated through the fourth notebook, as well as the set of tweets tagged during the last notebook using the RoBERTa model.
- **Wordcloud**. Folder with the images generated during the exploratory data analysis (fifth notebook).

Bibliography

[5] Shikah J. Alsunaidi et al. "Applications of Big Data Analytics to Control COVID-19 Pandemic". In: *Sensors* 21, 2282 (marzo 2021).

[9] Francesco Barbieri et al. "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1644–1650. DOI: 10.18653/v1/2020.findings-emnlp.148. URL: https://aclanthology.org/2020.findings-emnlp.148.

[11] Christian Beck. "Generalised information and entropy measures in physics". In: *Contemp. Phys* 50 (4) (2009), pp. 495–510.

[13] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research 3* 993-1022 (2003).

[15] Marco Bonzanini. *Mastering Social Media Mining with Python*. Packt, 2016.

[17] Siddhartha Chatterjee and Michal Krystyanczuk. *Python Social Media Analytics*. Packt, 2017.

[26] Anastasia Giachanou and Fabio Crestani. "Like it or not: A survey of Twitter sentiment analysis methods". In: *ACM Computing Surveys* 49, 2, artículo 28 (junio 2016).

[33] C.J. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI* Junio (2014).

[35] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Borrador (tercera versión), 2020.

[36] Ilker Kocabas, bekir Taner Dinçer, and Bahar Karaoglan. "Investigation of Luhn's claim on information retrieval". In: *Turkish Journal of Electrical Engineering and Computer Sciences* 19, Nº3 (2011).

[37] Sergei Koltcov. "Application of Rényi and Tsallis entropies to topic modeling optimization". In: *Physica A: Statistical Mechanics and its Applications* 512 (2018), pp. 1192–1204.

[44] Hugo A. Mitre-Hernández, Lemus-Olalde Cuauhtémoc, and Edgar Ortega-Martínez. "Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio". In: *Ingeniería Investigación y Tecnología* XV (número 3) (julio-septiembre 2014), pp. 403–418.

[53] Jesús Cordobés Puertas. *Tema 2 - El ambiente externo*. 2023.

[57] Ken Schwaber and Jeff Sutherland. *The Scrum Guide*. 2020.

[77] Xiaohui Yan et al. "A Biterm Topic Model for Short Texts". In: *Association for Computing Machinery* (2013).

Webgraphy

[1] *About The Nielsen Company*. Nielsen. Accessed on 18/05/2023. URL: https://web.archive.org/web/20090215003017/http://nielsen.com/about/index.html.

[2] *About us*. Nielsen. Accessed on 18/05/2023. URL: https://www.nielsen.com/about-us/about/.

[3] *About us*. Nielsen IQ. Accessed on 18/05/2023. URL: https://nielseniq.com/global/en/about-us/.

[4] María Fernanda Aguirre. *Realiza estimaciones ágiles y precisas gracias al Planning Poker*. appvizer. Accessed on 29/05/2023. URL: https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/planning-poker.

[6] *Anaconda Distribution: Free Download*. Anaconda. Accessed on 15/11/2022. URL: https://www.anaconda.com/download.

[7] Louie Andre. *53 Important Statistics About How Much Data Is Created Every Day*. Finances Online. Accessed on 19/05/2023. URL: https://financesonline.com/how-much-data-is-created-every-day/.

[8] David Arroyo. *Hi-Fi Rush costó la mitad que Forspoken y ya le supera en ventas*. Meristation. Accessed on 29/06/2023. URL: https://as.com/meristation/noticias/hi-fi-rush-costo-la-mitad-que-forspoken-y-ya-le-supera-en-ventas-n/.

[10] *Beautiful Soup Documentation*. Accessed on 19/03/2023. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[12] *Big Data en el fútbol: El caso Brentford FC*. acadef. Accessed on 05/06/2023. URL: https://www.acadef.es/big-data-en-el-futbol-el-caso-brentford-fc/.

[16] Jason Brownlee. *Difference Between a Batch and an Epoch in a Neural Network*. Machine Learning Mastery. Accessed on 15/11/2022. URL: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[18] Brian Dean. *How Many People Use Twitter in 2023?* backlinko. Accessed on 08/06/2023. URL: https://backlinko.com/twitter-users#twitter-users.

[19] Clément Delteil. *Unsupervised Sentiment Analysis With Real-World Data: 500,000 Tweets on Elon Musk*. Towards AI. Accessed on 12/04/2023. URL: https://pub.towardsai.net/unsupervised-sentiment-analysis-with-real-world-data-500-000-tweets-on-elon-musk-3f0653135558.

[20] *Developer Platform Documentation*. Twitter. Accessed on 02/02/2023. URL: https://developer.twitter.com/en/docs.

[21] Claire Drumond. *Guía de la metodología scrum: qué es, cómo funciona y cómo empezar*. TechCrunch. Accessed on 20/05/2023. URL: https://www.atlassian.com/es/agile/scrum.

[22] *error: Microsoft Visual C++ 14.0 or greater is required.* Microsoft. Accessed on 06/05/2022. URL: https://learn.microsoft.com/en-us/answers/questions/136595/error-microsoft-visual-c-14-0-or-greater-is-requir.

[23] Javier Escribano. *El rumor del lanzamiento de Advance Wars 1+2 en Switch esta semana es falso*. Hobby Consolas. Accessed on 13/04/2023. URL: https://www.hobbyconsolas.com/noticias/rumor-lanzamiento-advance-wars-12-switch-semana-falso-1196730.

[24] Matija Ferjan. *Xbox Game Pass Subscribers: How Many Game Pass Subscribers are There in 2023?* Headphones addict. Accessed on 04/06/2023. URL: https://headphonesaddict.com/xbox-game-pass-subscribers/.

[25] Jason Foster. *Data Skills Are Mission-Critical: How To Bridge The Skills Gap*. Forbes. Accessed on 04/06/2023. URL: https://www.forbes.com/sites/forbesbusinesscounci
2022/11/15/data-skills-are-mission-critical-how-to-bridge-the-skills-gap/.

[27] Enes Gokce. *NLP Capstone Project*. Accessed on 09/05/2023. URL: https://github.com/EnesGokceDS/Amazon_Reviews_NLP_Capstone_Project.

[28] Enes Gokce. *Sentiment Analysis on Amazon Reviews*. Towards Data Science. Accessed on 09/05/2023. URL: https://towardsdatascience.com/sentiment-analysis-on-amazon-reviews-45cd169447ac.

[29] David Gómez. *Social Media no traduce Redes Sociales*. bienpensado. Accessed on 19/05/2023. URL: https://bienpensado.com/que-es-social-media-y-su-diferencia-con-las-redes-sociales/.

[30] Aimee Hart. *Fire Emblem Engage hits 1.61 million sales worldwide*. Gayming. Accessed on 29/06/2023. URL: https://gaymingmag.com/2023/05/fire-emblem-engage-hits-1-61-million-sales-worldwide/.

[31] *How do you compare and contrast BERT with other deep learning approaches for sentiment analysis?* Linkedin. Accessed on 28/06/2023. URL: https://www.linkedin.com/advice/0/how-do-you-compare-contrast-bert-other-deep-learning.

[32] Owen Hughes. *Employers are desperate for data scientists as demand booms*. ZDNET. Accessed on 04/06/2023. URL: https://www.zdnet.com/article/employers-are-desperate-for-data-scientists-as-demand-booms/.

[34] Manish Singh Ivan Mehta. *Twitter to end free access to its API in Elon Musk's latest monetization push*. TechCrunch. Accessed on 02/02/2023. URL: https://techcrunch.com/2023/02/01/twitter-to-end-free-access-to-its-api/.

[38] Neville Lahiru. *Hi-Fi Rush Outperforms Forspoken in Steam Sales*. Gamerant. Accessed on 29/06/2023. URL: https://gamerant.com/hi-fi-rush-forspoken-steam-sales-outperformed/.

[39] Alejandro Manzanares Loreto. *¿Cuál es la diferencia entre Data Science y Data Analytics?* Hack a boss. Accessed on 02/06/2023. URL: https://www.hackaboss.com/blog/cual-es-la-diferencia-entre-data-science-y-data-analytics.

[40] Steven Loria. *TextBlob: Simplified Text Processing*. Accessed on 11/05/2023. URL: https://textblob.readthedocs.io/en/dev/.

[41] Fran G. Matas. *Las 21 consolas más vendidas de la historia*. Vandal. Accessed on 04/06/2023. URL: https://vandal.elespanol.com/reportaje/las-20-consolas-mas-vendidas-de-la-historia.

[42] Abby McCain. *26 STUNNING BIG DATA STATISTICS [2023]: MARKET SIZE, TRENDS, AND FACTS*. ZIPPIA. Accessed on 03/06/2023. URL: https://www.zippia.com/advice/big-data-statistics/.

[43] Rachel Meltzer. *These Are the Top Industries Hiring Data Analysts Right Now*. CF Blog. Accessed on 04/06/2023. URL: https://careerfoundry.com/en/blog/data-analytics/top-industries-hiring-data-professionals/.

[45] Andreas Mueller. *WordCloud for Python documentation*. Accessed on 12/04/2023. URL: https://amueller.github.io/word_cloud/.

[46] Fran Méndez. *¿Cómo el Big Data ayudó a Obama a ganar?* Forbes. Accessed on 05/06/2023. URL: https://forbes.es/start-ups/7560/como-el-big-data-ayudo-a-obama-a-ganar/.

[47] *NielsenIQ se convierte en una empresa independiente*. Business Wire. Accessed on 18/05/2023. URL: https://www.businesswire.com/news/home/20210308005771/es.

[48] *NielsenIQ's alternatives and competitors*. CBInsights. Accessed on 03/06/2023. URL: https://www.cbinsights.com/company/nielseniq/alternatives-competitors.

[49] *NLP Natural Language Processing : Introducción*. DataScientest. Accessed on 02/06/2023. URL: https://datascientest.com/es/nlp-introduccion.

[50] Matthias Orgler. *What is the optimal sprint length in Scrum?* Hackernoon. Accessed on 22/05/2023. URL: https://hackernoon.com/what-is-the-optimal-sprint-length-in-scrum-368e966f3243.

[51] Prasad Patil. *What is Exploratory Data Analysis?* Towards Data Science. Accessed on 16/06/2023. URL: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15.

[52] Martín Pellarolo. *metacritic*. Accessed on 17/03/2023. URL: https://github.com/martinpella/metacritic.

[54] Radim Rehurek. *Gensim*. Accessed on 05/02/2023. URL: https://pypi.org/project/gensim/#data.

[55] Shahmeer Sarfaraz. *GTA Online Update 1.66 Fixes Major Exploit Causing Player Bans*. Exputer. Accessed on 13/04/2023. URL: https://exputer.com/news/games/gta-online-update-1-66-exploit/.

[56] Bruce Schoenfeld. *El arma secreta del Liverpool: el análisis de datos*. The New York Times. Accessed on 05/06/2023. URL: https://www.nytimes.com/es/2019/05/29/espanol/liverpool-champions.html.

[58] *Story Point based Cost Estimation*. G DATA. Accessed on 02/06/2023. URL: https://www.gdatasoftware.com/blog/story-point-based-cost-estimation.

[59] Keith Stuart. *Interview: the science and art of Metacritic*. The Guardian. Accessed on 13/06/2023. URL: https://www.theguardian.com/technology/gamesblog/2008/jan/17/interviewtheartofmetacriti.

[60] *Sueldos para el puesto de Analista en España*. glassdoor. Accessed on 02/06/2023. URL: https://www.glassdoor.es/Sueldos/analista-sueldo-SRCH_KO0,8.htm?clickSource=searchBtn.

[61] *Sueldos para el puesto de Data Scientist en España*. glassdoor. Accessed on 02/06/2023. URL: https://www.glassdoor.es/Sueldos/data-scientist-sueldo-SRCH_KO0,14.htm?clickSource=searchBtn.

[62] Paul Tassi. *'Horizon Forbidden West: Burning Shores' Shows Metacritic Must Curb Review Bombing*. Forbes. Accessed on 13/06/2023. URL: https://www.forbes.com/sites/paultassi/2023/04/23/horizon-forbidden-west-burning-shores-shows-metacritic-must-curb-review-bombing/.

[63] Maksim Terpilowski. *Bitermplus*. Accessed on 09/05/2023. URL: https://bitermplus.readthedocs.io.

[64] Maksim Terpilowski. *Tmplot*. Accessed on 10/05/2023. URL: https://pypi.org/project/tmplot/.

[65] Mintu Tomar. *Xbox Smash Hit Hi-Fi Rush Proves Having a $70 Price Tag Is Not the Winning Formula for Modern Games*. Essentially Sports. Accessed on 29/06/2023. URL: https://www.essentiallysports.com/esports-news-xbox-smash-hit-hi-fi-rush-proves-having-a-70-price-tag-is-not-the-winning-formula-for-modern-games/.

[66] John Tones. *El cierre del estudio de 'Forspoken' deja claro algo: la industria actual del videojuego no perdona errores*. Xataka. Accessed on 29/06/2023. URL: https://www.xataka.com/videojuegos/cierre-estudio-responsable-forspoken-deja-clara-cosa-industria-actual-videojuego-no-perdona-errores.

[67] John Tones. *Microsoft compra Bethesda por 7.500 millones de dólares y se queda con franquicias como 'DOOM', 'Fallout' o 'Wolfenstein'*. Xataka. Accessed on 04/06/2023. URL: https://www.xataka.com/videojuegos/microsoft-da-empujon-a-su-cartera-exclusivos-compra-bethesda-editora-franquicias-como-doom-fallout-wolfenstein.

[68] Markus Tretzmüller. *Biterm*. Accessed on 08/05/2023. URL: https://pypi.org/project/biterm/.

[69] *Tweepy Documentation*. Accessed on 02/02/2023. URL: https://docs.tweepy.org/en/stable.

[70] *Twitter API Rate Limits*. Twitter. Accessed on 02/02/2023. URL: https://developer.twitter.com/en/docs/twitter-api/rate-limits.

[71] *Twitter API (subscriptions plans)*. Twitter. Accessed on 02/06/2023. URL: https://developer.twitter.com/en/products/twitter-api.

[72] *Twitter API v2 data dictionary*. Twitter. Accessed on 02/02/2023. URL: https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet.

[73] Rebekah Valentine. *Phil Spencer: Game Pass leads to more game sales*. Games Industry. Accessed on 29/06/2023. URL: https://www.gamesindustry.biz/phil-spencer-game-pass-leads-to-more-game-sales.

[74] Duong Vu. *Generating WordClouds in Python Tutorial*. Datacamp. Accessed on 12/04/2023. URL: https://www.datacamp.com/tutorial/wordcloud-python.

[75] *What is exploratory data analysis?* IBM. Accessed on 16/06/2023. URL: https://www.ibm.com/topics/exploratory-data-analysis.

[76] Leah J. Williams. *Square Enix says Forspoken sales were 'lacklustre'*. GAMES hub. Accessed on 29/06/2023. URL: https://www.gameshub.com/news/news/square-enix-forspoken-sales-lacklustre-2609303/.

[78] *¿Cuál es el coste de la empresa al contratar a un trabajador?* KENJO Blog. Accessed on 02/06/2023. URL: https://blog.kenjo.io/es/cual-es-el-coste-de-la-empresa-al-contratar-a-un-trabajador.