

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN  
INGENIERÍA DE TELECOMUNICACIÓN**

**TRABAJO FIN DE MASTER**

**DEVELOPMENT OF A TECHNOLOGICAL  
SURVEILLANCE TOOL AGAINST THE SPREAD OF  
TERRORIST IDEOLOGIES, BASED ON NATURAL  
LANGUAGE PROCESSING AND MACHINE LEARNING**

**ÁLVARO DÍAZ DEL MAZO**  
**2021/22**



## TRABAJO DE FIN DE MASTER

**Título:** DESARROLLO DE UNA HERRAMIENTA DE VIGILANCIA TECNOLÓGICA CONTRA LA DIFUSIÓN DE IDEOLOGÍAS TERRORISTAS, BASADA EN EL PROCESAMIENTO DEL LENGUAJE NATURAL Y EL APRENDIZAJE AUTOMÁTICO

**Título (inglés):** DEVELOPMENT OF A TECHNOLOGICAL SURVEILLANCE TOOL AGAINST THE SPREAD OF TERRORIST IDEOLOGIES, BASED ON NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

**Autor:** ÁLVARO DÍAZ DEL MAZO

**Tutor:** CARLOS ÁNGEL IGLESIAS

**Ponente:** PONENTE

**Departamento:** Departamento de Ingeniería de Sistemas Telemáticos

## MIEMBROS DEL TRIBUNAL CALIFICADOR

**Presidente:** —

**Vocal:** —

**Secretario:** —

**Suplente:** —

**FECHA DE LECTURA:**

**CALIFICACIÓN:**



**UNIVERSIDAD POLITÉCNICA DE MADRID**

ESCUELA TÉCNICA SUPERIOR DE  
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos  
Grupo de Sistemas Inteligentes



TRABAJO DE FIN DE MASTER

DEVELOPMENT OF A TECHNOLOGICAL  
SURVEILLANCE TOOL AGAINST THE SPREAD OF  
TERRORIST IDEOLOGIES, BASED ON NATURAL  
LANGUAGE PROCESSING AND MACHINE LEARNING

JULY, 2022



# Resumen

---

Conforme evoluciona la tecnología, son más las posibilidades de divulgar información de manera rápida y barata. Multitud de periódicos o revistas online, son accesibles hoy en día a través de internet, de manera global. La facilidad con la que se transmite la información, trae asociada algunos riesgos ligados a la “desinformación”. Un claro ejemplo donde se aprecia este hecho, es en el reclutamiento terrorista, especialmente de la mano del Estado Islámico (EI).

Tras años de lucha contra este movimiento radical en Europa, no termina de radicarse el problema. Después de los numerosos ataques asociados a este grupo terrorista, es necesario el desarrollo de nuevas herramientas para contrarrestar la expansión. La tecnología, es una de las claves que habilitan dichos desarrollos.

Esta tesis consiste en la creación de una herramienta de clasificación automática de artículos o noticias, publicados en diferentes canales de comunicación web (principalmente periódicos), en contra de la propagación de ideologías radicales **islamistas**. Para ello, el trabajo se ha dividido en dos partes.

Una primera parte en la que se han desarrollado dos modelos basados en técnicas de aprendizaje automático y procesamiento de lenguaje natural, mediante los cuales es posible realizar la clasificación. Ambos modelos se han comparado, para finalmente utilizar el que genera mejores resultados.

La segunda parte ha consistido en la implementación de una cadena de tareas que permite extraer noticias y artículos de diferentes canales web. Posteriormente, estos artículos son procesados y clasificados (se aplica el modelo desarrollado). Finalmente se almacenan en una base de datos y se representan en una interfaz gráfica que permite visualizar distintos resultados.

**Palabras clave:** Aprendizaje automático, Procesamiento del lenguaje natural, Propaganda radical, Estado Islámico, Python



# Abstract

---

As technology evolves, there are more and more possibilities to disseminate information quickly and cheaply. Many online newspapers and magazines are now globally accessible via the internet. The ease with which information is transmitted brings with it some risks linked to “misinformation”. A clear example of this, is terrorist recruitment, especially by the Islamic State (IS).

After years of fighting this radical movement in Europe, the problem has not yet been solved. After the numerous attacks associated with this terrorist group, it is necessary to develop new tools to counter its expansion. Technology is one of the key enablers of such developments.

This thesis consists of the creation of a tool for the automatic classification of articles or news, published in different web communication channels (mainly newspapers), against the propagation of radical **Islamist** ideologies. To this end, the work has been divided into two parts.

A first part in which two models based on machine learning and natural language processing techniques have been developed, by means of which it is possible to carry out the classification. Both models have been compared, in order to finally use the one that generates the best results.

The second part consisted of the implementation of a pipeline that allows the extraction of news and articles from different web channels. Afterwards, these articles are processed and classified (the developed model is applied). Finally, they are stored in a database and represented in a graphical interface that allows the visualisation of different results.

**Keywords:** Machine Learning, Natural Language Processing, Radical propaganda, Islamic State, Python



# Agradecimientos

---

Me gustaría agradecer especialmente a todas las personas que me han ayudado y apoyado durante la realización de este proyecto, y en especial a aquellas que lo han hecho de más primera mano como han podido ser mi tutor, Carlos Ángel Iglesias, que me ha guiado y apoyado desde el comienzo hasta el final; y a otros compañeros del GSI como pueden ser Guillermo García, Jorde Tardío, Óscar Araque y Jaime Conde. Sin duda, su ayuda ha sido fundamental para sacar adelante la tesis. Tampoco me puedo olvidar de otros compañeros que me han animado y apoyado, así como a mi familia.

Mi más sincero agradecimiento a todos ellos.



# Acknowledgement

---

I would like to greatly thanks to all the people that have supported and helped me during the project development. I would like to thanks especially to those who have done it at first hand, such as my tutor, Carlos Ángel Iglesias, who has guided and supported me from the beginning to the end; and to other GSI colleagues such as Guillermo García, Jorde Tardío, Óscar Araque and Jaime Conde. Undoubtedly, their help has been essential to carry out the thesis. I can not neither forget about other colleagues that have supported and animated, as to my family.

My sincere thanks to all of them.



# Contents

---

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Agradecimientos</b>	<b>XI</b>
<b>Acknowledgement</b>	<b>XIII</b>
<b>Contents</b>	<b>XV</b>
<b>List of Figures</b>	<b>XIX</b>
<b>List of Tables</b>	<b>XX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Project goals . . . . .	2
1.3 Structure of this document . . . . .	3
<b>2 State of Art</b>	<b>5</b>
2.1 Data creation, collection and exploitation. Evolution and importance. . . . .	5
2.2 Information and social media evolution . . . . .	11
2.3 Islamic state. History, expansion and attacks. . . . .	17
2.4 Radical text detection and classification over online media . . . . .	19
<b>3 Enabling Technolgies</b>	<b>23</b>
3.1 Machine Learning and Artificial Intelligence . . . . .	23
3.1.1 Balanced Data . . . . .	26
3.1.1.1 Over Sampling . . . . .	27
3.1.1.2 Under Sampling . . . . .	28
3.1.1.3 Class Weight . . . . .	29
3.1.1.4 Threshold . . . . .	29
3.1.2 Types of Machine Learning . . . . .	29

3.1.2.1	Supervised Learning . . . . .	30
3.1.2.2	Unsupervised Learning . . . . .	32
3.1.2.3	Reinforcement Learning . . . . .	34
3.1.2.4	Deep Learning . . . . .	36
3.1.3	Types of algorithm . . . . .	38
3.1.3.1	Decision Trees . . . . .	40
3.1.3.2	Logistic Regression . . . . .	41
3.1.3.3	K-Nearest Neighbours . . . . .	42
3.1.3.4	Support Vector Machine (SVM) . . . . .	43
3.1.3.5	Naïve Bayes . . . . .	44
3.2	Natural Language Processing . . . . .	45
3.2.1	Pandas . . . . .	45
3.2.2	NLTK . . . . .	46
3.2.3	Stylomepy . . . . .	47
3.2.4	Imblearn . . . . .	47
3.2.5	GSITK . . . . .	47
3.2.6	Gensim . . . . .	48
3.3	Evaluation Techniques . . . . .	48
3.3.1	Accuracy . . . . .	49
3.3.2	Precision . . . . .	49
3.3.3	Recall . . . . .	50
3.3.4	F1 Score . . . . .	50
3.3.5	Cross Validation . . . . .	50
3.3.6	Grid Search CV . . . . .	51
3.4	Dashboard technologies . . . . .	51
3.4.1	GSI Crawler . . . . .	52
3.4.2	Elasticsearch . . . . .	52
3.4.3	Graphical interface . . . . .	52
3.4.4	Luigi . . . . .	53
<b>4</b>	<b>Architecture and Methodology</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Data Extraction . . . . .	57
4.3	Data Processing . . . . .	60
4.4	Data Storage . . . . .	63
4.5	Data Visualization . . . . .	65

<b>5</b>	<b>Classification Model and Evaluation</b>	<b>67</b>
5.1	Data used . . . . .	68
5.2	Model 1 - Similarity-based deep-learning model . . . . .	72
5.2.1	Pre-processing of the data . . . . .	72
5.2.2	Data Analysis . . . . .	73
5.2.3	Feature extraction . . . . .	76
5.2.4	Classification and evaluation . . . . .	78
5.3	Model 2 - Stylometry . . . . .	81
5.3.1	Preprocessing of data . . . . .	81
5.3.2	Metrics calculation . . . . .	82
5.3.2.1	Readability Index . . . . .	82
5.3.2.2	Vocabulary Richness . . . . .	84
5.3.2.3	Formality measure . . . . .	86
5.3.3	Data Analysis . . . . .	88
5.3.4	Classification and Evaluation . . . . .	90
5.4	Evaluation . . . . .	94
<b>6</b>	<b>Conclusions</b>	<b>97</b>
6.1	Conclusions . . . . .	97
6.2	Achieved Goals . . . . .	99
<b>A</b>	<b>Impact of this project</b>	<b>101</b>
A.1	Social impact . . . . .	101
A.2	Economic impact . . . . .	102
A.3	Environmental impact . . . . .	102
A.4	Ethical impact . . . . .	102
<b>B</b>	<b>Economic budget</b>	<b>105</b>
B.1	Project structure . . . . .	105
B.2	Physical resources . . . . .	107
B.3	Human resources . . . . .	107
B.4	Conclusion . . . . .	108
	<b>Bibliography</b>	<b>110</b>



## List of Figures

---

2.1	Growth of online data creation throughout history [79] . . . . .	6
2.2	Types of data [48] . . . . .	8
2.3	Data life-cycle approaches . . . . .	10
2.4	Big Data. The 5Vs . . . . .	11
2.5	Information Media evolution [12] . . . . .	13
2.6	Social media timeline [84] . . . . .	14
2.7	The three layers of the Internet [59] . . . . .	16
3.1	Balanced and Imbalanced Datasets [82] . . . . .	27
3.2	Example of imbalanced dataset in python . . . . .	28
3.3	Supervised Learning Diagram [37] . . . . .	31
3.4	Unsupervised Learning Diagram . . . . .	33
3.5	Reinforcement Learning diagram [32] . . . . .	35
3.6	Deep Learning - Neural Network [31] . . . . .	36
3.7	Evaluation Techniques confusion matrix [52] . . . . .	49
3.8	Evaluation Techniques - cross validation [76] . . . . .	51
4.1	Pipeline architecture diagram . . . . .	56
4.2	Luigi user interface . . . . .	57
4.3	Count of articles scraped by date . . . . .	59
4.4	Count of articles scraped by date - grouped by source . . . . .	59
4.5	Example of retrieved articles features . . . . .	60
4.6	Most common words in articles grouped by prediction . . . . .	61
4.7	Most common words in headline grouped by prediction . . . . .	62
4.8	Total distribution of radical and non radical articles . . . . .	62
4.9	Number of radical and not radical articles by source . . . . .	63
4.10	Objects stored in Elasticsearch . . . . .	64
4.11	Dashboard Kibana . . . . .	65
5.1	Radical Magazine Dataframe . . . . .	69
5.2	Proppy training Dataframe . . . . .	70

5.3	Dataframe label proportion before balancing data . . . . .	71
5.4	Histogram features Similarity-based DL model . . . . .	74
5.5	Features correlation Similarity-based DL model . . . . .	75
5.6	Histogram number of sentences - Similarity-based DL model . . . . .	76
5.7	Algorithms used in the models . . . . .	78
5.8	Best classifier for Similarity-based DL model . . . . .	79
5.9	F-Score typical values [39] . . . . .	88
5.10	Metrics results Stylometry model . . . . .	88
5.11	Histogram Metrics Model Stylometry . . . . .	89
5.12	Histogram Metrics (Normalized) Model Stylometry . . . . .	90
5.13	Metrics correlation Model Stylometry . . . . .	91
5.14	Histogram MATTR - Model Stylometry (label = 0) . . . . .	92
5.15	Histogram MATTR - Model Stylometry (label = 1) . . . . .	92
5.16	Best classifier for Stylometry Model . . . . .	93
5.17	Common words by source (radical articles) . . . . .	96
5.18	Top 5 sources with more radical articles . . . . .	96

# List of Tables

---

3.1	Comparison between SL and UL [7]	34
3.2	Stemming vs Lemmatization	47
4.1	Possible endpoints for the GSI Crawler	58
5.1	Balancing methods Accuracy	72
5.2	Similarity-based DL model results	80
5.3	Similarity-based DL model results using 67% of training data	81
5.4	Length for each dataframe in Stylometry model	82
5.5	ARI Scores related to US grade level	83
5.6	Fog Index related to US grade level	84
5.7	Flesch Index related to US grade level	85
5.8	Adjective scores - categories and typical values	87
5.9	Model Stylometry results (trained with 80% of the data)	94
5.10	Comparison of models' results	96
6.1	Conclusion - comparison of models' results	99
6.2	Summarized results of the models	100
B.1	Project structure	106
B.2	Human resources costs	108
B.3	Total costs of the project	109



# Introduction

---

*This chapter is going to introduce the context of the project, including a brief overview of all the different parts that will be discussed in the project. It will also break down a series of objectives to be carried out during the realization of the project. Moreover, it will introduce the structure of the document with an overview of each chapter.*

## 1.1 Context

For some decades, the growth of digital communication has meant the opening of new digital media channels, in which, the identity of the communicator cannot be assured, the number of people reached by the message, exceeds exponentially the number reached by traditional media such as television or radio, and the message can be distributed without any kind of filter or control. Moreover, it is distributed free of charge in most cases. This scenario is real, and it is happening nowadays. That is why, in recent years, these channels have been used by different groups or organisations to disseminate ideas and knowledge. It is true that sometimes these ideas favour the development of well-being and contribute to society. But, on the other hand, there are also negative uses, such as the propagation of radical ideas. This is precisely the case study of this thesis, which focuses on the **detection of radical propaganda** texts, specifically for **Islamic ideology**.

For this purpose, the thesis is based on two main parts. The first one consists of the **development of two natural language processing models**. Each module is based

on the extraction of a different type of features. In this way, it is possible to check which features are more favourable for text classification. The first of the models works with **Similarity** [8], which is the process of determining the intention behind a series of words by looking the similarity with a set of predefined features. Based on this classification, a prediction is made as to whether the text is considered radical or not. The second model works with the **stylometry** [21] of the text. Stylometry is a set of measures, based on the way the text is written, that allow us to identify authors and classify texts. Again, based on the classification, the prediction is made.

The second part of the project has consisted in the **design, implementation and testing of a pipeline** that allows the visualization of a set of graphs that collect the results of all the texts analyzed. This process starts with the extraction of news and articles from several online channels (some newspapers and Google news). Then the articles are processed, analyzed, and features are extracted, according to the requirements of each model. Finally, they are stored in elasticsearch, a database that feeds the web interface where the graphs to be presented have been defined.

This project introduces some changes compared to other similar projects of the intelligent systems group. In particular, the extraction of texts from American newspapers such as the New York Times and CNN, from radical magazines such as Dabiq and Rumiya, and the novelty of Google News is introduced. In addition, the possibility of applying customized models to the pipeline, instead of using pre-existing fixed models, is explored. This feature greatly favours the growth of natural language processing, as it allows for a quick visualization of the results obtained after applying a custom model to a set of texts. In addition, it favours also the fight against radicalism, and more specifically, Islamic radicalism.

## 1.2 Project goals

In this section there are presented the main goals identified before the thesis started, and it defines the minimum set of requirements that need to be matched in order to consider the thesis as finished. At the conclusion of the thesis, there will be a comment about the achievement or not of the goals presented below. Therefore, the main goals of the project are the following:

- G1 - Deepen the use of natural language processing and machine learning techniques. These tools are increasingly used in professional environments for the creation of intelligent bots, task automation or even real-time translators. That is why specialization, in terms of development and implementation, in such techniques will be useful for professional development.

- G2 - Develop at least one Natural Language Processing model which takes as input English texts, and predict if the text is radical with at least an 85% of score if the text comes from the same source, and, at least, a 70% of score if the text comes from a different source.
- G3 - Modify the existing GSI Pipeline to include customized models instead of a general one. Modify also the input of the pipeline, to include other channels like Google News.
- G4 - Promote the awareness of all those people who may be targeted by radical organizations and promote the continuous fighting against radical ideologies.

With the achievement of these tasks, it will be also simplified the monotonous work of online propaganda analysts and inspectors, who are looking for signs of radical ideologies. By creating this filtering tool, it is not intended to replace the human specialists dedicated to these tasks, but rather to simplify their work, which sometimes becomes monotonous, heavy, and exhausting. In short, it is a way to optimize human effort in searching better results in the identification of radical propaganda.

## 1.3 Structure of this document

In this section it is provided a brief overview about the chapters that are included in this thesis. The complete explanation and description of each chapter, can be found in the corresponding section. The structure is as follows:

**Chapter 1 - Introduction.** It presents an introduction of the project. Main goals are explained as well as the motivations for doing this thesis.

**Chapter 2 - State of Art.** It describes the State of Art. It is divided into four sections that covers the overall situation around the project. These sections are the data growth (it remarks the importance of online data filtering), the way social media has evolved (it remarks the importance of online surveillance), the evolution the Islamic State has had (it remarks the importance of centring the efforts on this topic), and the existence of other text analysis methods (it establishes a basis for this project).

**Chapter 3 - Enabling technologies.** The enabling technologies that have been used in the project are described in this section. For both, the model's development and the implementation of the pipeline and the dashboard, the technologies are described deeply. On the one hand, for the model development part, it is provided a general overview of the current technology situation, presenting also some of the most used algorithms. Also, the main modules that have been used and the evaluation techniques are described. On the second hand, for the dashboard part, there are presented the main modules that are needed

for the development and well-function of the site. In this case, a more technical description is provided in the specific section for this matter.

**Chapter 4 - Architecture.** It describes the architecture followed to implement the pipeline and therefore, to build up the dashboard. It contains a more technical description of each part of the pipeline.

**Chapter 5 - Classification models.** This section analyses deeper the processing module of the pipeline, which consist in the models developed. Both models are presented, including the different parts of the development. Finally, an evaluation of the models is done.

## State of Art

---

*In this section, the state of art is presented. Firstly, in section 2.1, a general overview about how the technology evolution has affected to the data growth during the last years. Section 2.2 explains how the media has evolved and how is this evolution related with the data evolution. In section 2.3, the point is set on how the Islamic State has evolved and how their activities are affecting people around the world. Lastly, the section 2.4, which contains a more technical point of view, establishes a framework about online and social media radical text detection and classification.*

### **2.1 Data creation, collection and exploitation. Evolution and importance.**

Since prehistoric times, technology (in every sense of the word) has accompanied the development of human beings, the way they act towards their environment and the way they act towards each other. From the discovery of fire and the days when weapons were made from sticks and stones, through the development of medicine with the creation of anesthesia, to the present day, every step that has been taken in technology has had a great influence on the way we live and work. According to the definition of technology, “*Set of theories and techniques that enable the practical use of scientific knowledge*” [70], it can be said that each of these eras has been marked by the use or application of a technology. Starting in

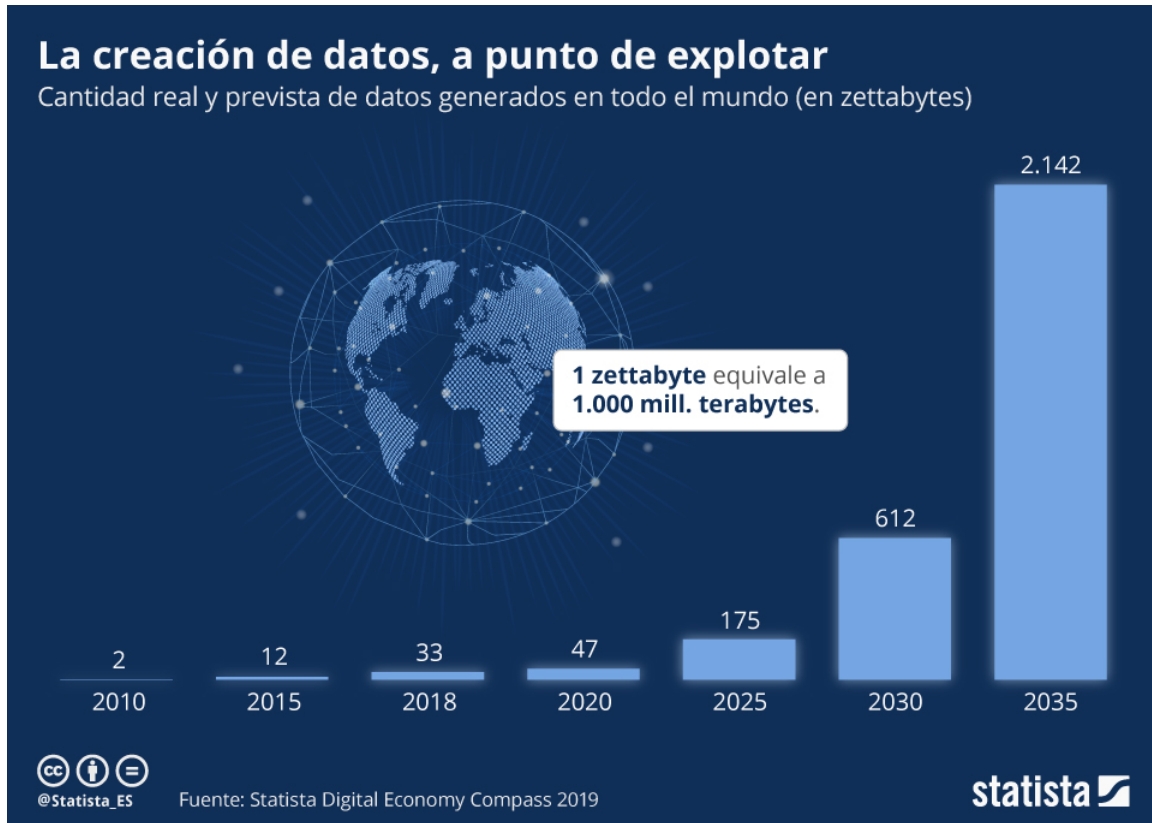


Figure 2.1: Growth of online data creation throughout history [79]

prehistoric times with the use of fire as a means of survival and passing through the Middle Ages with the use of gunpowder as a means of defence, we have reached the contemporary age, where the technology that is determining the way of life and the one that is considered as the most valuable asset, is data. For more than a decade, the world has been experiencing exponential growth in the volume of data flowing through the network, and nowadays, data is taken from everything not only digital but also physical, as the fingerprints, the facial recognition of mobiles and the steps someone does each day, are physical data transformed into digital and then stored somewhere. According to a study by Statista [79] the volume of data generated in 2018 reached 33 zettabytes<sup>1</sup>, which multiplies the value by 16.5 compared to the previous decade. This can be seen in the Figure 2.1.

There have been identified some factors that have contributed to this growth during the last decades. These factors are:

- The growth in the number of smart devices connected to the internet. Wearables, Smart houses, and Smart vehicles are clear examples of what these kind of devices mean. The famous American enterprise, Gartner, forecasted that the worldwide end-

<sup>1</sup>1 zettabyte is equivalent to 1000 millions of terabytes

user spending on wearable devices will increase 18,1 points in 2021 respect to the year before [34]. Currently, most people around the world, use their intelligent devices to do daily activities like reading the newspaper, ordering a supermarket shopping, or just chatting with the friends.

- The fact commented before is strongly supported by the continuous evolution of cloud technologies. The Big Five or GAFAM<sup>2</sup> are currently leading this sector, although Amazon stands out from the others with an estimated rate of 32 per cent of the business [78], thanks to the service provided by Amazon Web Services (AWS). It is far followed by Microsoft azure with a 19 per cent of the business and then by Google Cloud with a 7 cent of the business.
- Last but not least, the evolution of new generation of communications. The advent of 5G is a reality, and an increasing number of antennas are supporting these communication networks, providing greater mobile network coverage around the world. The fundamental pillars of 5G communications are the Connectivity (Massive IoT), which allows to connect up to 1 million of low-power nodes per square kilometre, the Throughput and capacity (Enhance mobile broadband), which is expected to faster up to 100 times current download and upload speeds, and Latency, which will decrease end-to-end cellular network latency to 5 ms or less. Therefore, 5G technology is data service driven, as the main services it offers use lots of data.

All these factors, among others, are the reason of the increasing volume of data travelling through internet. However, this fact also brings negative aspects. The more information through the net, the more difficult to detect negative content (fake news, radical thoughts, racist tendencies, etc.).

In addition to the ethical differentiation of data, data can be further divided into input and output data. On the one hand, input data is the one provided by users, and on the other hand, output data is the one provided by machines. When this division arose, the difference was that computers could not calculate anything, nor generate output data, without prior user's input. Over time, this situation has changed due to the **storage of information**.

In today's situation, data is not only generated by people, but also by digital sources based on measurements of the environment or previous data records. Weather satellites are an example that has been with us for many years. In addition to these systems, more modern systems such as digital sensors are also helping the evolution towards Industry 4.0.

Due to the big amount of data sources, it is needed a first broadly categorization of data into the following types [48] as shown in Figure 2.2:

---

<sup>2</sup>GAFAM: Google, Amazon, Facebook, Apple and Microsoft

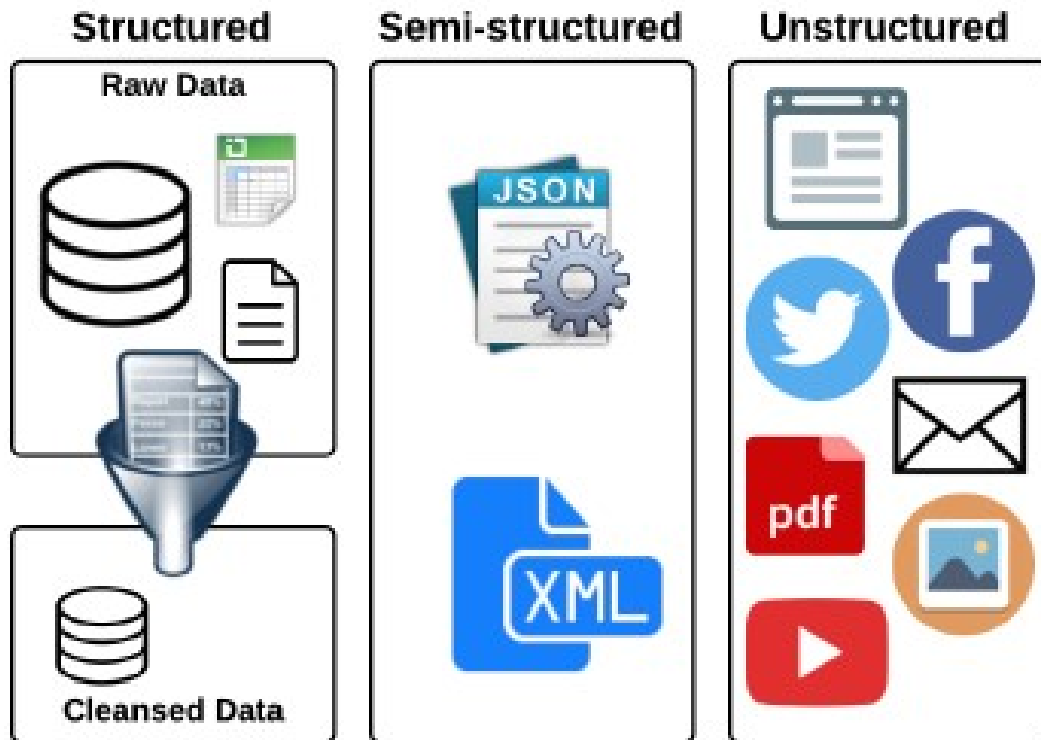


Figure 2.2: Types of data [48]

- **Structured data.** It is data conventionally captured in business applications in the form of data residing in a relational or non-relational database. This kind of data is also categorized in “raw data” and “cleaned data”. Data that is taken in as it is, without much cleansing or filtering, is called raw data. Data that is taken in with a lot of cleansing and filtering, catering to a particular analysis by business users, is called cleansed data.
- **Unstructured data.** All the other data, which doesn’t fall in the category of structured, can be called unstructured data. Data collected in the form of videos, images, and so on are examples of unstructured data. The main current sources that generate this kind of data are social media.
- **Semi-structured data.** This category has come into existence because of the Internet and is becoming more and more predominant with the evolution of social sites. Some of the examples of semi-structured data are the well-known data formats, namely JavaScript Object Notation (JSON) and Extensible Markup Language (XML).

With the growth and the diversification of data, different storage needs have arisen. The way in which data is stored has also been a niche market over the last few years. Some time

ago, since data was not the main source of value for companies, there was not too much concern about how data was stored. It was simply enough to have an ordinary database where only the crucial information was stored. Over time it has become clear that this approach was no longer valid and that data brings even more value than the people or assets themselves.

This new approach has brought with it other ways of storage. Among these new storage models, it can be highlighted the data lakes and the data warehouses.

On the one hand, a data Lake can be defined as a vast repository of a variety of organization(s), raw information that can be acquired, processed, analysed and delivered. A Data Lake acquires data from multiple sources in one or more organizations in its native form and may also have internal, modelled forms of this same data for various purposes. The information thus handled could be any type of information, ranging from structured or semi-structured data to completely unstructured data. A Data Lake is expected to be able to derive relevant meanings and insights from this information using various analysis and machine learning algorithms.

On the other hand, a data warehouse is a data repository where data is organized in such a way that all the elements related to an event or an object, are stored together. The data are time sensitive, as the changes produced are recorded to generate reports.

Actually, the main differences between these two concepts are the stage when data is pre-processed. In data lakes, all types of data are stored together and without keeping the same format. Therefore, the processing of the data is made after the storing, so you can define different processing methods, depending on the final needs for what you want the data. For the data warehouses, data is processed before storing, to keep the format and the grouping of the data. By this way, all the data is ordered before storing and some applications benefit from this approach.

However, this both approaches are not mutually exclusive and can be used together. In this sense, the approach starts storing data in a data lake and then creates multiples data warehouses as the result of the different processing methods applied to the data lake data. Although this approach needs much more infrastructure, it is very useful for big companies that operate in different markets with many applications. The three different approaches are represented in the Figure 2.3.

One of the technologies that has wrapped and pushed the trends in this direction has been Big Data. Big data [48] [43] [91] is a term for data sets so large or complex that traditional data processing applications are inadequate to handle them. The term “big data” often simply refers to the use of predictive analytics, user behaviour analytics or some other advanced data analysis methods that extract value from data. Generally, big data then refers to large volumes of hard-to-manage data that can overwhelm businesses on

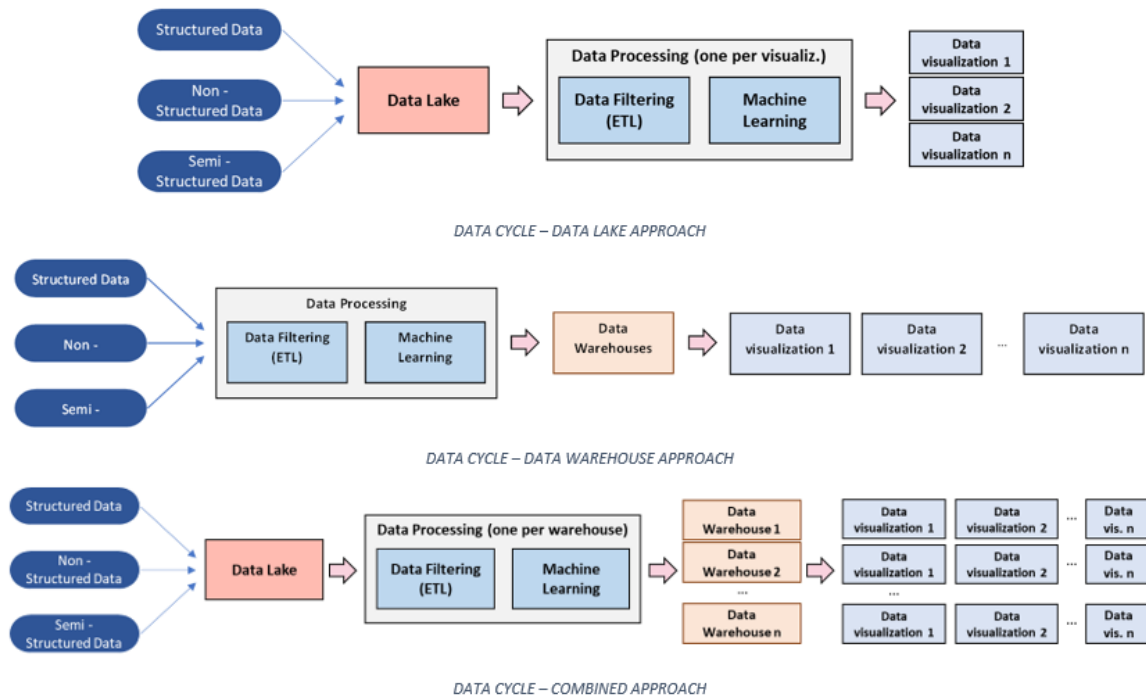


Figure 2.3: Data life-cycle approaches

a day-to-day basis. The data and the processes carried out with it provide companies with knowledge on the basis of which decisions can be improved and give confidence to make strategic moves.

Whenever you encountered the term big data being overly used, you must have come across an important aspect with regard to it, called 5 “V” (until recently, it was 4Vs, and even before 3Vs). The 5Vs, namely variety, velocity, volume, variability and veracity (in no particular order) determine whether the data we call Big Data really qualifies to be called “big”:

- Variety refers to vivid types of data and the myriad sources from which these are arrived at. With the proliferation of technologies and the ever-growing number of applications, there is high emphasis on data variety. Broadly, data types can be categorized as it was defined above (Figure 2.2).
- Velocity is referred in two aspects in the concept of big data. First is the rate at which the data is generated, and second is the capability by which the enormous amount of data can be analysed in real time to derive some meaningful conclusions. This “V” makes it easy to take quick decisions in real time.
- Volume refers to the amount/scale of data that needs to be analysed for a meaningful result to be derived. There isn’t a quantitative figure that categorizes a data to

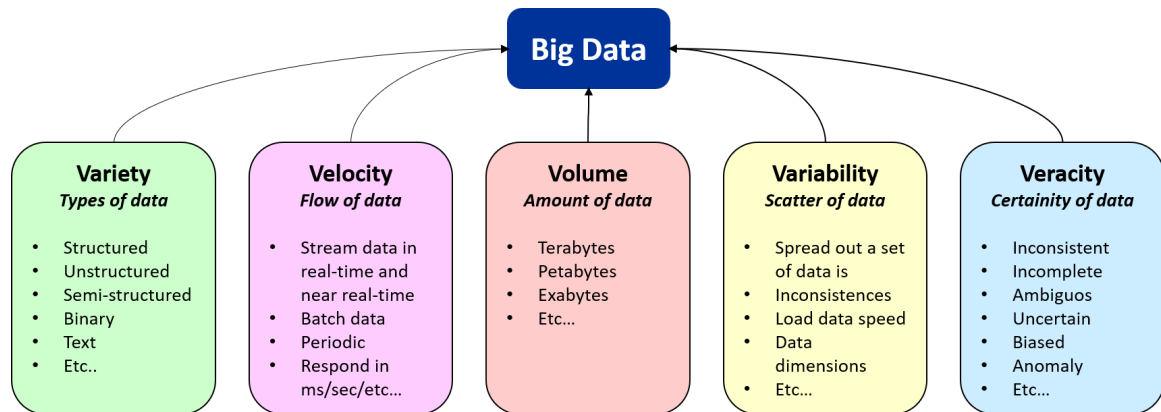


Figure 2.4: Big Data. The 5Vs

be falling into big data. But usually, this volume is definitely more than what a conventional application is handling as of now. So, in general, this is quite big and does pose a problem for a traditional application to deal with in a day-to-day fashion. Both application users and the applications themselves are producing tons of data daily in the form of conventional transactions and other analytics. The first of the factors discussed at the beginning, which deals with wearables and smart devices, could be included in this “V”.

- Variability refers to a few different things. It refers to how spread out a set of data is, to the number of inconsistencies in the data or to the inconsistent speed at which big data is loaded into your database. Big data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
- Veracity refers to accuracy of data being analysed to get to a meaningful result. With a variety of sources, especially the not-so-reliable user-entered unstructured data, the data coming from some of these channels has to be consumed in a judicial manner. The fake news/materials that circulate on social media, on the internet and in the media every time a relevant event happens around the world are already well known. This V brings up a very important aspect of accuracy in big data. With the proliferation of data, especially on social channels, this “V” is going to be very important.

## 2.2 Information and social media evolution

Taking a brief look at the history of information media, we have to go back to the 15th century, when the printing press was invented and replaced manuscripts. It is needed to remember that previously, all the information that was disseminated was through manuscripts,

mainly supported by the church. The idea of the printing press was conceived by Johann Gutenberg around 1450 and transformed the dissemination of knowledge in Europe. As a result, books began to emerge, which expanded the possibilities of communication and the dissemination of reading and writing. Until the 17th century, it was the production of books in different languages that allowed knowledge and information to spread across different territories. From that time on wards, with the publication of newspapers, mainly in Western Europe, new methods of communication emerged that allowed much more up-to-date and adapted information to be provided.

From the publication of the first newspapers until the beginning of the 20th century, this medium, together with magazines and books, became the main source of information for the population. This form of communication brought about changes in the way people acted and felt, as it made much more feasible to share news over hundreds of kilometres. News that could appear in America could reach Europe in a matter of days or weeks, which, for that time, meant great advances and great changes. At the level of mentality, it was a fundamental factor for the population to expand its horizons.

The effectiveness of the printed letter was very convincing, and it was unrivalled until the appearance of other mass media, which again meant a new rupture in the way of communicating. This factor occurred during the first decade of the 20th century, and it was none other than the improvement of radio equipment, which became lighter. Thanks to these advances, radiotelephony became widespread and allowed transmissions over long distances to be greatly accelerated.

The trend continued in the following years, when the first television broadcasting was achieved. Over the next few years, thanks to the development of technology and telecommunications, the first public broadcasts were launched. Although television was not very accessible to many families because of the price, over the years it became more widespread and it began to gain importance as a mean of communication. The launching into orbit of the first communication satellites is a fact that favours the extension of television as a mean of communication.

During the remainder of the 20th century, television became the most widely used medium of communication, thanks to its ease of transmission and reach. It was at the end of the 20th century that Arpanet appeared, considered to be the birth of the Internet [10] [83]. Year after year, the number of internet users around the world is increasing exponentially. Several companies were beginning to see the potential of digital content, and this growth means that the internet was starting to become a new mean of communication through which to find all kinds of news on demand. Fig. 2.5 represents the evolution of communications from the Gutenberg's printing press to social media [12].

With the advent of the internet and during its first years of growth, the first social

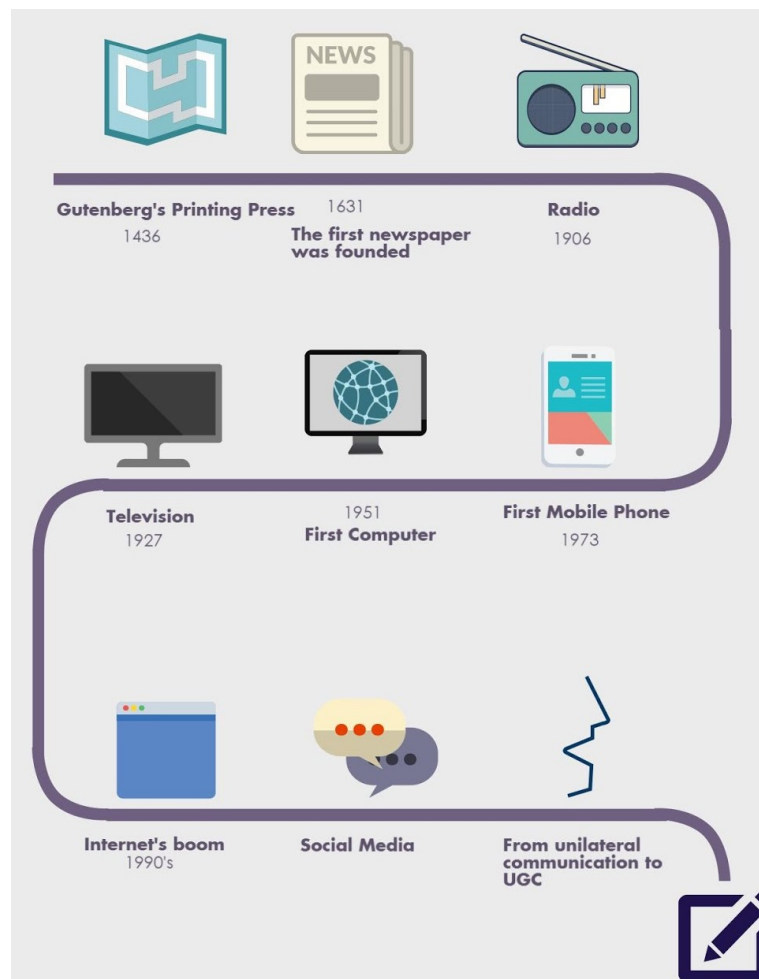


Figure 2.5: Information Media evolution [12]

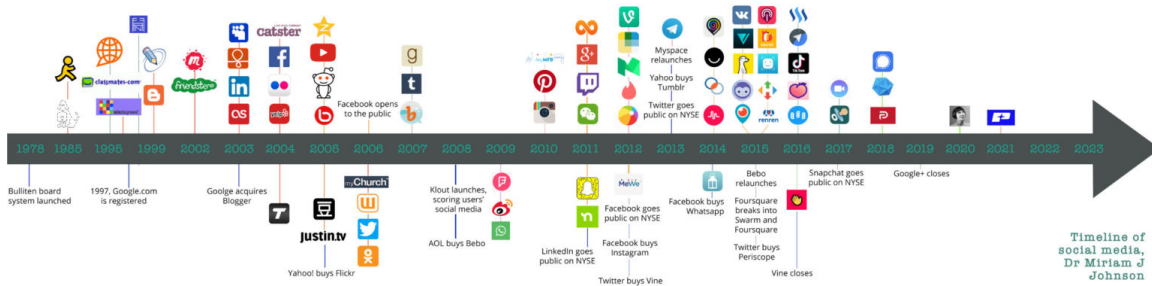


Figure 2.6: Social media timeline [84]

networks began to grow [83]. Between the ‘90s and the ‘00s, the internet’s growth enabled the introduction of online communication services such as CompuServe, America Online, and Prodigy. They introduced users to digital communication through email, bulletin board messaging, and real-time online chatting.

This gave rise to the earliest social media networks, beginning with the short-lived Six Degrees profile uploading service in 1997. This service was followed by many others during the next years, and all these platforms actually attracted millions of users, what enabled email address registration and basic online networking.

Nowadays, it can be highlighted some of the applications launched during the last century, like LinkedIn, which was launched in 2002 and now it has grown more than 675 million of users; Facebook, which was launched in 2004 and has more than 2.8 billion of monthly users [36]; Twitter, which was founded in 2006 and has more than 396 million users (most popular users aged is between 25 and 34) [22] [40]; and TikTok, which was founded in 2016 and has reached 1.2 billion monthly active users in Q4 2021 and is expected to reach 1.5 billion by the end of 2022 [45]. The Figure 2.6 tries to sum up the launch date of almost all the social media applications for the last 30 years [84] [49].

The statistics reflect the latest step in the evolution that has taken place, not only in technology and social media, but also in the way we communicate and interact with audiences. Social media has become the main source of data today and is largely categorised as “unstructured data”.

The evolution of technology has transformed interpersonal relationships into the digital world, where the integrity of users cannot be verified<sup>3</sup>. The ease of access to any of these platforms, and the ability of large companies such as Google or Facebook to collect and store personal data, can be detrimental or harmful to users.

Personal data is defined as “*any information which are related to an identified or identifiable natural person by name, numerical identifiers, online identifiers, location, or, one or more physical, psychological, genetic, mental, economic, cultural or social characteris-*

<sup>3</sup>integrity: to be certain that a user is who they say he/she is

*tics*” [65], and is protected at European level by the General Data Protection Regulation (GDPR) [66]. However, due to misinformation and improved cyber-attacks, it is becoming increasingly common to fall victim to phishing, fraud and other attacks on personal identity.

Nevertheless, these are not the only attacks that need to be addressed, as there are many other misuses of social media and the internet in general. These misuses include the sale of illegal weapons, the sale of drugs or the indoctrination of civilians by radical groups.

The exponential growth of data, the globalization of the internet and the hidden face of the network (Darknet and Deep web), makes it increasingly difficult to identify this type of event. One of the main challenges lies precisely in identifying possible attacks. To explain this concept properly, it is worth defining the different access layers of the web [29] [59].

First, there is the “Surface web”, which is the internet that everyone uses and surfs. It is a network on which we are easily traceable through our IP address. It is mainly made up of the pages indexed by conventional search engines such as Google, Wikipedia or Yahoo, but also of all those other websites that can be accessed publicly even if they are not indexed, such as Facebook, Twitter and other social networks, websites or blogs. In any case, the data that circulates at this level is only a small part of the data that navigates the entire web.

Secondly, there is the “Deep Web”, which is the opposite of the previous level. At this level, more than 90% of the data that exists on the internet can be found, although it is not possible to access it publicly. This layer is characterized by the presence of private pages (protected by passwords), such as all files stored in infrastructure providers like Google or Microsoft, or pages protected by a paywall, and by the presence of non-indexable pages, which are all those that are generated on a temporary basis as a result of a search on a travel website or after making banking queries.

And finally, the Dark-net. This is not a layer as such, but is included within the “Deep Web” and constitutes less than 1% of its content. To access this area of the web, it is necessary to use specific networks such as TOR or I2P. The inner workings of Tor are designed to camouflage where we are coming from and where we are going, so this part of the web is characterized by the exposure of mainly [28], illegal content. From the sale of weapons and drugs, to the recruitment of assassins and child pornography, these are examples of what can be found on the Dark-net.

These explanation is better shown in the Figure 2.7, which represents metaphorically the different levels and the contents that can be found in each one.

On the other hand to what has been commented, there is other content that is not only visible on the dark web, but also on the public internet and even on social networks. That is, the indoctrination of citizens based on radical ideologies, such as Islam. The internet has become the main mean of communication, and as such, it is the main channel used to

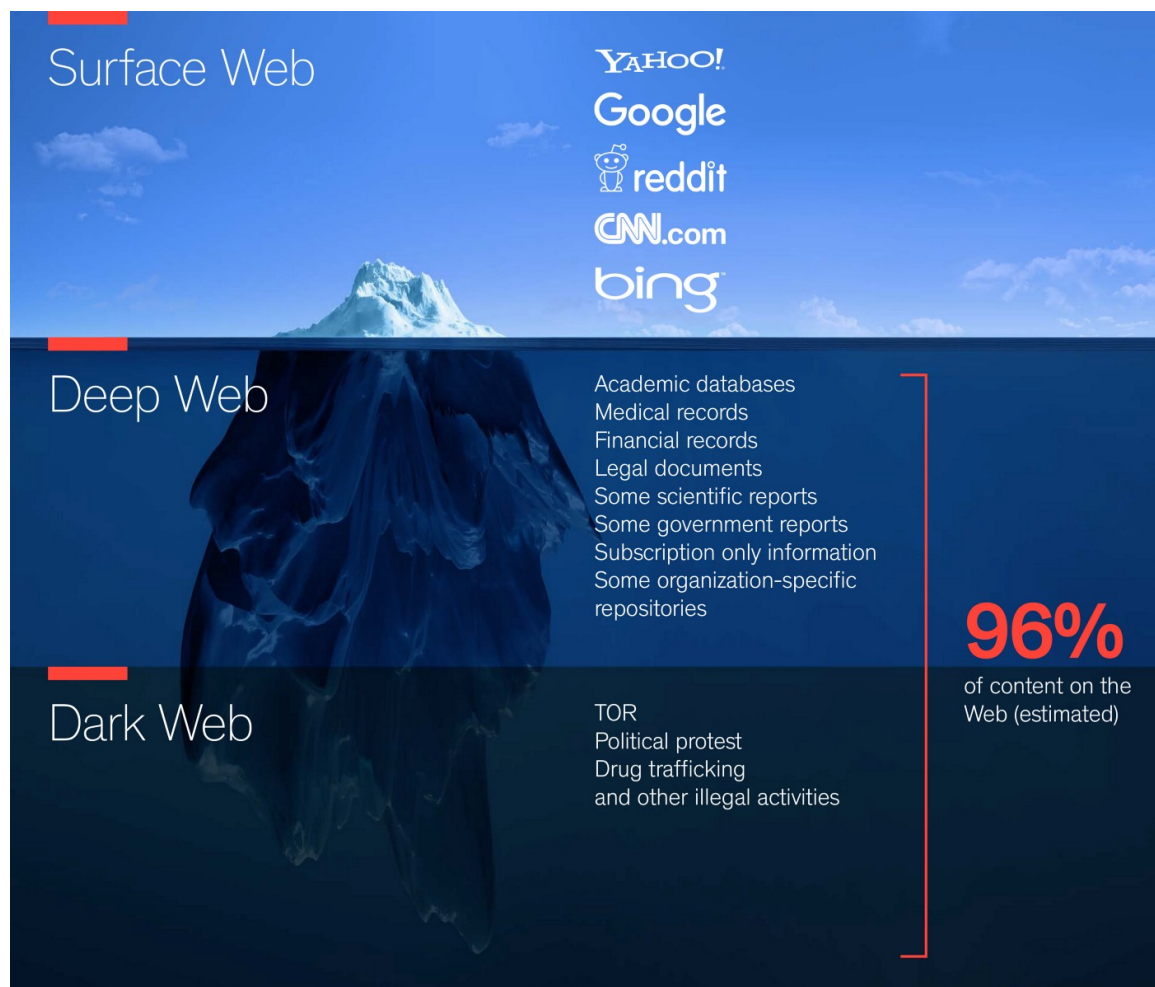


Figure 2.7: The three layers of the Internet [59]

promote radical tendencies as it is a very effective tool for terrorists and radical groups to spread propaganda. One of the main problems is to detect these radical messages in order to block them or promote counter-narratives.

Terrorist organizations have used different media strategies. According to New York Times [74], Islamic State of Iraq and Syria (ISIS) media strategy can be named Jihad 3.0 since they follow a sophisticated multidimensional strategy [3] [27], including not only social media (e.g., Twitter) but also video games [3], online magazines, like Dabiq and Rumiya, high-quality videos, audio reports in SoundCloud or publication of battle summaries in JustPaste, to name a few [21].

Therefore, it is very important to continue working on the development of tools or methods to detect online terrorist propaganda and disinformation, in order to continue the fight against Islamic radicalization.

## 2.3 Islamic state. History, expansion and attacks.

To define the evolution of the Islamic state, it is first necessary to define the concept of ideology. Ideology, according to the RAE (Real Spanish Academy), is defined as [69]:

1. *Conjunto de ideas fundamentales que caracteriza el pensamiento de una persona, colectividad o época, de un movimiento cultural, religioso o político, etc.*
2. *Doctrina que, a finales del siglo XVIII y principios del XIX, tuvo por objeto el estudio de las ideas.*

According to these definitions, more and more ideologies exist in today's society. Some clear examples of this are the different political-cultural strands, beyond what has always been known as "left" and "right", feminism, anarchism, etc. Since it is not possible to go through all of them, this section will focus on the Islamic religion [68]

Islam emerged during the 6-7th century with the Prophet Muhammad (Mahoma). After his death, two strands emerged in the quest for power. These were Sunniism and Shi'ism, which clashed over who had the legitimate right to lead Muslims. After many years, these clashes continue to this day, although the majority presence of the Sunni current means that they are the ones who impose their beliefs. As is normal, throughout history, this majority current has evolved up to the present day, where three main types of religious movements can be distinguished [94]:

- Moderate Islamism.
- Salafist movement.

- Militant Islamism or Jihadism.

It will be the last of these that will have the greatest impact on this thesis, since it is from this movement that the ideas and initiatives to recruit new militants come from.

Jihadism can be further subdivided into two strands. On the one hand, Qutbism, which justifies the use of violence to further Islamist goals (overthrowing the current state regime). On the other hand, Salafi-jihadism, which emerges as a violent evolution of Qutbism combined with Salafist movements, refers to the ideology that **actively promotes and conducts** violence and terrorism in order to achieve the establishment of an Islamic state or a new caliphate [99] [35].

Salafi-jihadism fulfils the four defining functions of an ideology and should be treated as such, not as a religion:

- It promotes the image of Islam as an object of attack by non-Muslims, and Western 'crusaders' in particular.
- Blames Europe for using colonization and diplomacy to prevent their economic rise and relegate their culture.
- It promotes a superior and intolerant Muslim-soldier identity.
- Advocates certain actions, such as the adoption of Sharia law and acts of terror.

Furthermore, it also matches the definition of radicalization in accordance with [14], which says: "*Radicalization is a process through which people become increasingly motivated to use violent means against members of an out-group or symbolic targets to achieve behavioural change and political goals*". The radicalisation model is summarised in three stages. First one is the sensitive phase, which consists of becoming sensitive to radical ideas. Then, the membership stage, which consists of becoming part of a radical group. Lastly, the action phase, which consists of acting on behalf of the radical group's ideologies.

As discussed in section 2.2, the evolution of technology and the growth of the internet has opened up the possibility of communication through new channels based on the internet and social media. As a consequence, the internet has been one of the most widely used means to recruit not only attackers, but also online intermediaries to carry messages and disseminate martyr videos.

The main threats of recruitment through the internet are due to the wide dissemination of extremist ideas, and are mainly those listed below [14]:

- Recruitment of ISIS fighters and searching for lone wolf terrorists online.
- Rapid dissemination of extremist ideologies.

- Identification of outraged, unemployed, anti-nationalist youth online that can be easily influenced.
- Many other jihadi groups have started adopting ISIS's method of mobilizing supporters globally through social media.
- The establishment of online jihadi community may accelerate more rapidly with increasing global radicalization.
- Greater penetration of terror groups into social media accounts to target youth.
- Online donations and fund-raising activities are becoming more digital and ease to reach masses over social media makes such activities spread easily.

Salafi-jihadism has involved some well-known terrorist groups such as al-Qaeda or Daesh, which have been behind several attacks in recent years, most notably the tragic 11-S attack in the United States. This attack was carried out by the al-Qaeda group, and is associated with the Islamist ideology of Salafi-jihadism. It is perhaps one of the most remembered attacks because of the consequences it had, but like this one, there are many others that have taken place in recent years, and which force us to consider these radical groups as great dangers to society. These attacks include, first, those developed by the radical al-Qaeda terrorist group [103], and secondly, those carried out by ISIS [95].

This growth over the last few years has put the Islamic State in the spotlight, and as a consequence, has led to the development of tools to fight radicalization and indoctrination, especially online.

## 2.4 Radical text detection and classification over online media

Based on the situation established in the previous sections of the state-of-the-art (section 2.1, section 2.2, section 2.3), and going out of the historical context, a study has been carried out on the possible paths to follow in order to fulfil the objectives established in the project (Section 1.2).

The research has focused on the detection of radical Islamic State propaganda. One of the first possible detection techniques identified is the use of stylometry [21]. The paper says as follows, *"The Internet has become an effective tool for terrorist and radical groups to spread their propaganda. One of the current problems is to detect these radical messages in order to block them or promote counter-narratives"*. [9] The use of stylometry is based on the study of the form of language, and more specifically on the study of:

- Function words: frequency of various function words.

- Frequent words: frequency of most frequent words.
- Punctuation: frequency of punctuation characters<sup>4</sup>.
- Hashtags: frequency of most frequent hashtags.
- Letter bigrams: frequency of most frequent letter bigrams.
- Words bigrams: frequency of most frequent word bigrams.

From the above aspects, the study of hashtags stands out, as it is the only one that does not strictly pertain to the form of language. This is because this study collects the data source from Twitter. As discussed in section 2.1, the growth of the data is exponential, and data filtering is necessary in order to homogenize the data. To do this, as indicated in the paper [9], the 10 most used hashtags are: IS, AllEyesOnISIS, Iraq, Syria, Islam, ISIS, Muslims, Mosul, Caliphate, and Khilafa. On the other hand, for the other thesis [21] the metrics that were used are the following:

- Readability index: is a style metric that measures how easy or difficult reading a text is.
- Vocabulary Richness: Measures the lexical diversity of the texts.
- Formality metrics: measures the degree of formality of a text.
- Coherence Measure: is an index that evaluates the coherence of a text.

Another work related to stylometry is presented in [63], which also serves as a basis for the one mentioned above [9]. For this paper, the use of stylometry is also applied to the identification of the author of a text, and, as it indicates, *“The primary application of author identification during this period shifted from literary attribution to forensics — identifying authors of terrorist threats, harassing messages, etc. In the forensic context, the length of the text to be classified is typically far shorter and the known and unknown texts might be drawn from very different domains. Some forensic linguists who focus on legal admissibility scorn the use of online corpora for training of classifiers due to concerns over the level of confidence in the ground truth labels”*.

The study of stylometry has a strong influence not only on the language, since the way of writing and expressions are very much influenced by it, but also on the different ways in which it is used in different parts of the world. For example, the English spoken in the USA differs from the English spoken in the UK, even though it is the mother tongue in

---

<sup>4</sup>Punctuation characters: . , ; : ' , - [ ] , , ! , ? , &

both countries. The same is true, even on a larger scale, in other countries where English is not even the mother tongue. This topic has been researched in [88], where the usefulness of the study of stylometry for the identification of authors, texts and radical propaganda is corroborated.

Besides, the possibility of using other techniques has been explored, as presented in the paper [51], where Deep Neural Networks are used for detecting radicalization on online media. This paper presents a study to detect radicalisation through online social media, using text analytics and deep learning mechanisms to identify radical context. The mentioned approach consists on the following steps:

1. Data preparation: includes data collection (from blogs, articles, tweets, etc.), data annotation (defining if the text is radical or non-radical) and data preprocessing (cleaning punctuations, blank texts, stopwords, etc.).
2. Cohen's Kappa Coefficient: is a measure of inter-expert agreement between two experts/raters on our annotated data.
3. Training and visualization.
  - (a) Generating **Word-Embeddings**. All the word embedding representation methods are based on a hypothesis that words generally carry similar meaning that occur in similar contexts.
  - (b) **LSTM (Long short-term memory) and Fully-connected Layer**. A feed-forward neural network supported by LSTM is used for weight training as it is presuming texts to be interrelated.
  - (c) Trend analysis. The results of classification are visualized graphically.

The results obtained with this approach (Feedforward NN supported by LSTM) are great in terms of precision (85.96%), but not in terms of recall (53.26%) and F-score (65.77%). Anyway, the results are better than using a simple classifier.

Finally, another technique commonly used for all kinds of analysis based on natural language processing (NLP) is sentiment analysis, which, in many cases can be reduced to a binary classification to reduce the complexity of the model. A clear example where this technique is used to classify radical texts is [2]. This article is focused on the problem of classifying a tweet as extremist or non-extremist. In addition, it states that the task of extremist affiliation detection can be reduced to a binary classification task. However, it also proposes and investigates the use of deep learning-based sentiment analysis techniques, as it has already shown a very good performance across in problems like vision and speech analytics. The article also tries to solve two questions.

On the one hand, how to recognize and classify extremist and non-extremist texts. The answer proposed by the article is using deep-learning-based sentiment analysis technique, namely LSTM + CNN. By applying this method, very good results were obtained. These results were even better than the ones from the previous research, in terms of accuracy (92.66%), precision (88.32%), recall (89.47%) and F-measure (90.71%).

On the other hand, how to perform sentiment classification of user opinions in relation to the emotional affiliations of extremists on Twitter and the Deep Web (this concept was presented in section 2.2). To do so, the author of the paper proposes a personalized classifier, based on emotion classification of user comments in relation to the affiliation of extremists on the Dark Web. This model also outperforms supervised machine learning classifiers in terms of accuracy, precision, recall and F-measure.

## Enabling Technologies

---

*This chapter offers a brief review of the main technologies that have made possible this project, as well as some of the related published works. All the technologies that have been used along the work are free to use, unless some libraries or references (specially books) that were not strictly necessary to finish the thesis. Besides, the order of the technologies has a lot of sense, because it goes from the more general technologies to the specific ones for this work. For each of the technologies, a briefly description will be written to understand the need of the technology in this work and how is applied. Firstly, the section 3.1 describes the concept of Machine Learning (ML) and Artificial Intelligence (AI), as well as some key ideas like the different types of learning that could be used to train a machine, or the algorithms that are typically used in machine learning models. Then, the section 3.2 describes the concept of Natural Language Processing, which has been the main enabler of the thesis, and mention some of the principal libraries that have been used. After it, the section 3.3 describes briefly the evaluation techniques that have been used to measure the performance of the models. Finally, in the section 3.4, it is presented the technologies used to develop the dashboard that are used to represent the results of the project.*

### 3.1 Machine Learning and Artificial Intelligence

Artificial Intelligence (AI) and Machine Learning (ML) are two fields very related that have evolved a lot during the last years. In fact, there are many people who think that both

concepts are the same. However, despite both concepts are related, there are not the same. AI is a bigger field that includes ML in addition to other technologies [6], like pattern recognition, data mining, computer vision, etc.

Despite Machine Learning is included in Artificial Intelligence, it is still a big field. Gartner defines AI [64] as applying advanced analysis and logic-based techniques, including machine learning, to interpret events, support and automate decisions, and take action.

Such is the growth that AI has experienced in recent years that even Gartner says it will be the main driver of decision making in the coming years [33]. They estimate that AI projects in place will be doubled for the leading organizations of the sector in the incoming years. On the other hand, this fact affects negatively also small companies, because they don't have the enough capability to invest in development of new trends like competitive algorithms of ML.

This problem is visible today, as almost any automation process is (or tends to be) carried out through AI techniques, as they facilitate monotonous tasks and speed up many of these processes (sending personalized advertising, answering frequently asked questions, estimating sales, spam detection, etc).

There are so many applications in which AI is applied, that it is divided into two types [24]. These are the weak and the strong AI. The main difference between them is the ability to learn from the decisions already taken and the consequences of those actions. Basically, weak AI is only able to perform the actions for which it was pre-programmed, and strong AI is pre-programmed with more complex algorithms that helps it to act in different situations. Therefore, the strong AI acts more like a human brain and enable to perform more difficult tasks (in fact, the strong AI is in the core of almost all robotic devices).

Nowadays, it exists some methods to classify an AI system between weak and strong. The first one most known is the Turing Test, which concept was introduced by Alan Turing, and was extended some years later by Harnard [38]. The extension done by Harnard demands the system not only the capabilities of Natural Language Processing (NLP), Automated reasoning, Knowledge representation and Machine Learning, but also Computer Vision and Robotics.

The Turing Test was criticized by John Searle, who said that a system passing the Turing test doesn't mean to be intelligent. It is not only necessary for a system seem to be intelligent, but also it needs to be. This idea was explained in the article [73] with the example of the Searle's Chinese room.

As the years went by, the requirements for a system to be considered intelligent have been more restrict. The current requirements demand the system, among others, to be build up with intelligent mechanisms, to matches or exceed human intelligence and to have real conscious minds. On the other hand, an AI system is considered weak just by simulating

human cognition.

Some examples of strong AI are the Searle's Chinese Room, speech recognition like virtual agents (these ones need to understand what humans try to say), and image recognition (which is mostly driven by self-driving applications and by the military world). On the other hand, some examples of weak AI could be self-driving cars (where Google and Elon Musk are the pioneers), spam detection, cross selling or credit card fraud detection.

For the case of this thesis, which is the development of a tool to filter out negative propaganda, it is used the weak AI, because it is based in some static rules that need to be satisfied to detect the patterns. In addition, those rules need to be preconfigured based on a set of examples with a known output about if it is or not negative propaganda.

According to Arthur Samuel [18], "*ML is a field of study that gives computers the ability to learn without being explicitly programmed*". This definition was extended some years later by Tom M. Mitchell in his book Machine Learning, where he said that [62] "*ML is not more than a field of study that gives the ability to learn without being explicitly programmed, or said in another way, a computer program that learns from an experience ( $E$ ) with respect to some class of task ( $T$ ) and performance measure ( $P$ ), if its performance at tasks in  $T$ , as measured by  $P$ , improves with  $E$* ".

Therefore, Machine Learning focuses the effort in the use of data and algorithms to face many kinds of problems in the same way humans do. The development of the algorithms is fully based on mathematics and statistics and will be out of the scope of this thesis. However, the algorithms used will be overall explained. Algorithms are used to detect and to learn patterns. After learning, the model has the possibility to classify and make predictions. The information produced as an output of the model is used to make decisions reason-based.

The steps that enable a model to work and make predictions are the following [77]:

1. **Collecting data:** Despite being the first step to build up a ML mode, is one of the most important ones. Data collection is sometimes a monotonous task so the aim for which the data is being collected could be easily lost. When it is about collecting data for a specific application goal, is not so easy to find valid and efficient data. It is of utmost importance to collect reliable data so that the ML model can find the right patterns and structures. The quality of the data that is used to feed the model will determine the accuracy of the model.
2. **Preparing the data:** This step is considered the filter to select the most useful data among all the one collected. This step presents some requirements for the data like for example to be distributed and unbiased, to be homogeneous, and others. After the cleaning part, is very useful to represent and visualize the data to understand

how it is structured and to see at a glance possible relationship between variables or classes. In addition, and depending on the model and the amount of data previously collected, it is also possible to split the data into two sets (training and testing).

3. **Choosing a model:** the model to be chosen is highly dependent on the application for which it is required. The models are fully based on mathematics, so for many years, scientists and engineers have worked on the development of various models suited for different tasks (image recognition, spam detection, etc.).
4. **Training the model:** The prepared data is passed to the model chosen to make predictions or find patterns. The model learns from the data how to perform the task set. Despite thinking that the more amount of data for training, the better performance, actually the model can be over trained. It is necessary to find the right proportion of data to train a model.
5. **Evaluating the model:** This step is the output of the one before. After training the model, it is necessary to see the performance it has with the data that were set aside for testing <sup>1</sup>. To evaluate the results obtained, several metrics could be used. For this thesis there will be used (at least) the accuracy (Section 3.3.1) and the f-score (Section 3.3.4).
6. **Parameter tuning:** this step consists on analysing the parameters that have been used for the training to see if they can be improved in such a way. There are existing ways for this task like for example grid search (Section 3.3.6).
7. **Making predictions:** As a final step, predictions can now be made with an accuracy that depends on the previous steps. The aim is to always aim for the highest possible accuracy. However, this task can become more complicated the more complicated the problem is.

### 3.1.1 Balanced Data

Machine Learning is so dependent on the data that is used to train the models. The data needs to be unbiased and needs to cover the most possible cases for what the final application is used. Furthermore, the data needs to be (almost) balanced [82], what means that the samples need to be uniformly distributed. The Figure 3.1 shows what do this mean.

The main problem of imbalanced data is that the model is going to learn that the best performance is to predict always the predominant class. In the Figure 3.1, predicting

---

<sup>1</sup>If the test were performed on the same data used for training, the output would be unrealistic, as the model is already used to that data and would find the same patterns in it as during the training phase. This will give a disproportionate accuracy

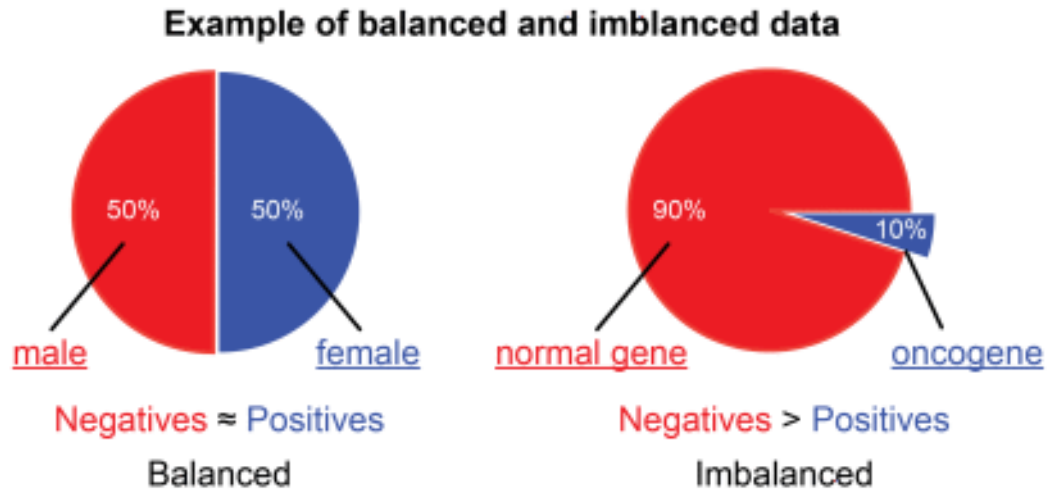


Figure 3.1: Balanced and Imbalanced Datasets [82]

always a normal gene will have a 90% of accuracy. This means that the result doesn't change depending of the input data, but the output will be always the one that matches with the predominant class.

It exists some methods to work with imbalanced datasets [82]) in python. These methods can be used by importing the “imbalanced-learning” package [58]. For example, considering an imbalanced data set with the distribution of the figure 3.2, it can be seen that the number of red samples surpasses the number of green samples. If a model is trained using this data, it will probably doesn't depend of the input and it will always predict the class 0. Therefore, it is necessary to balance the data. For this purpose, it can be found the following methods.

#### 3.1.1.1 Over Sampling

Oversampling [23] is a technique which increases the number of samples of the smallest class up to the size of the biggest class. This is done by generating synthetic samples. The advantage of oversampling is that no information is lost from the original training set, as all observations of the minority and majority classes are retained. On the other hand, it is prone to overfitting. Different techniques can be applied to oversample a class, but for this thesis there are presented two of them.

1. Random Over sampler [55]. The samples are randomly created taking into account the existing ones. This makes the number of samples of both classes equal, without adding any new features, as the samples added are the same as the existing ones. For this method, it is necessary to set up, among others, a parameter called “random state” [50]. This parameter controls the randomization of the algorithm, so fixing this

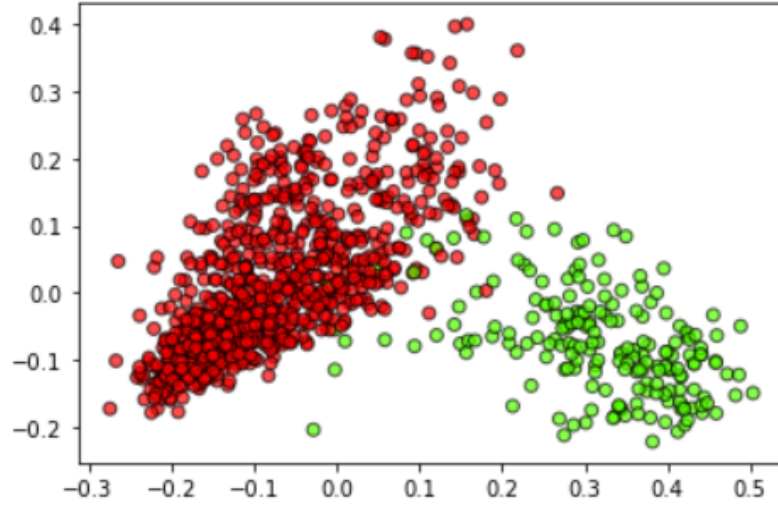


Figure 3.2: Example of imbalanced dataset in python

parameter will provide always the same output of randomization.

2. SMOTE (Synthetic Minority Over-sampling Technique) [57]. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. This method creates new samples with different features. For this method, it is necessary to configure, among other, the parameter “k-neighbours”, which represents the number of nearest neighbours used to construct synthetic samples.

### 3.1.1.2 Under Sampling

Undersampling [23] is a technique which decreases the number of samples of the biggest class down to the size of the smallest class. This is done by removing some samplers from the biggest class. The main difference with the Oversampling, is that some features could be lost during the removal of samples, and therefore some useful information might be discarded. On the other hand, it has other advantages like improving the run-time of the methods by decreasing the amount of training data, and helping in solving the memory problems. In such a way to the oversampling, different techniques can be applied to undersample a class, but for this thesis are presented two of them.

1. Random Under Sampler [56]. The samples are randomly removed. Since it is removing observations from the original data set, it might discard useful information. In the same way to the Random Over Sampler, the parameter “random state” needs to be configured.

2. NearMiss [54]. When two points belonging to different classes are very close to each other in the distribution, this algorithm eliminates the data-point of the larger class thereby trying to balance the distribution. NearMiss adds some heuristic rules to select samples. These rules are based on nearest neighbours algorithm. Despite this method also allows some parameters, for this thesis none of them have been configured (they keep the default value).

### 3.1.1.3 Class Weight

Setting the class weight constitutes another valid alternative for balancing [23]. Each scikit-learn classification model can be configured with a parameter, called “class weight”, which receives the weight of each class in the form of a Python dictionary. In order to calculate the weight of each class, it can be set the weight of the biggest class to 1 and set the weight of the smallest class to the ratio between the number of samples of the biggest class and the number of samples of the smallest class. By this method the number of samples for each class is not modified, but each sample is assigned with a normalized weight.

### 3.1.1.4 Threshold

Adjusting threshold [16] is a manual technique to balance a dataset. Conceptually, if a predicted value is greater than the threshold, it is set to 1, otherwise, it is set to 0. For those classification problems that have a severe class imbalance, the default threshold can result in poor performance. As such, a simple and straightforward approach to improving the performance of a classifier that predicts probabilities on an imbalanced classification problem is to tune the threshold used to map probabilities to class labels. There are many different ways to calculate the threshold, but for this thesis it will be used the value of the threshold which maximizes Youden’s J statistic [90], which formula is in Equation 3.1.

$$J = \frac{\text{truepositives}}{\text{truepositives} + \text{falsenegatives}} + \frac{\text{truenegatives}}{\text{truenegatives} + \text{falsespositives}} - 1 \quad (3.1)$$

The threshold obtained as result of the previous operation (Equation 3.1) is 1, what means that the best threshold behaves as the imbalanced case. Therefore, this balance method will be discarded for the development of the thesis.

## 3.1.2 Types of Machine Learning

Once the collected data has been processed and is ready to be used, it is time to choose a model. For this phase, predefined models are already available, but it is also possible to create new models by combining existing ones, which is known as a “Pipeline”. The model chosen must be suitable for each application and for the type of training to be carried

out. Models can learn in different ways: Supervised Learning, Unsupervised Learning, Reinforcement Learning, etc.

Similarly, there are different types of tasks and models, which, together with the characteristics of the data, make up the ingredients of ML. It is also necessary to take into account the type of problem to which the model has to respond. In this sense, problems will be distinguished as continuous or discrete (categorical). Models can be grouped in the following way:

- Descriptive model: studies the problem and tries to answer the question “What has happened?” It may include a diagnostic part where the reason is also studied.
- Predictive model: as its name suggests, this model attempts to make a prediction of what will happen.
- Prescriptive model: it studies how to make something happen. This is the most difficult phase, but the one that provides the most value.

In the following sub-sections, the types of training are described and related to the different types of models and problems, to identify the algorithms and applications for which they should be used.

### 3.1.2.1 Supervised Learning

Supervised Learning (SL) [60] is a machine learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. As the output is regarded as the label of the input data or the supervision, an input-output training sample is also called labelled training data, or supervised data. Occasionally, it is also referred to as Learning with a Teacher, Learning from Labelled Data, or Inductive Machine Learning. The goal of supervised learning is to build an artificial system that can learn the mapping between the input and the output, and can predict the output of the system given new inputs. If the output takes a finite set of discrete values that indicate the class labels of the input, the learned mapping leads to the classification of the input data. If the output takes continuous values, it leads to a regression of the input. The input - output relationship information is frequently represented with learning-model parameters. When these parameters are not directly available from training samples, a learning system needs to go through an estimation process to obtain these parameters.

Figure 3.3 represents the diagram of the workflow of a supervised learning model [37], which reveals that the quality, diversity and volume of the dataset is the real advantage between one model and another. The accuracy of the model depends primarily on the quantity and quality of the underlying dataset.

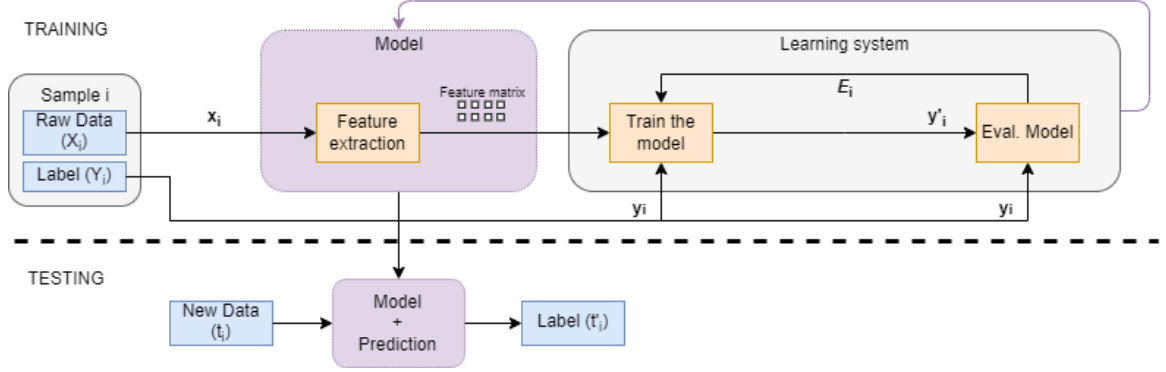


Figure 3.3: Supervised Learning Diagram [37]

The Figure 3.3 shows a diagram that illustrates the form of Supervised Learning. In this diagram, the raw data (input;  $X_i$ ) and the labels (output;  $Y_i$ ) represent the supervised training sample, and the “i” is the index of the training sample. During a Supervised Learning process, a training input  $X_i$  is fed to the Learning System, and the Learning System generates an output  $\hat{Y}_i$ . The Learning System output  $\hat{Y}_i$  is then compared with the ground truth labelling  $Y_i$  to compute the difference between them. The difference, termed Error Signal, and represented in this diagram with  $E_i$ , is then sent to the Learning System for adjusting the parameters of the learner. The goal of this learning process is to obtain a set of optimal Learning System parameters that can minimize the differences between  $\hat{Y}_i$  and  $Y_i$  for all “i”, i.e., minimizing the total error over the entire training data set.

The main advantage of supervised learning is that all the outputs generated/modified by the algorithm are meaningful to humans, so it is very helpful for discriminate pattern classification and data regression. On the other hand, it also has some disadvantages such as the difficulty or work involved in labelling each of the inputs with which the system will be trained. It must be taken into account that in some situations, labelling has a subjective factor, so there may be uncertainties or contradictions in the supervision.

Since it is a human-guided learning mode, the main applications for which it is used are to simulate human performance. The ability of machines to process inputs and calculate outputs makes it possible to perform tasks faster and more accurately than humans. In contrast, they are much more limited in complex tasks for which they have not been specifically trained. The applications are divided into two blocks.

Supervised Learning has been successfully used with categorical data to solve classification problems by the utilization of some algorithms as K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Naïve Bayes and induction trees. Medical Imaging is an example of this ingredients.

In addition, Supervised Learning has been also successfully used with continuous data

to solve regression problems (linear, logistic and polynomial) by the utilization of decision trees and random forests. The house price prediction could be an applicative example of this ingredients.

Supervised Learning [100] is also applied in other areas such as Information Retrieval, Data Mining, Computer Vision, Speech Recognition, Spam Detection, Bioinformatics, Cheminformatics, and Market Analysis.

### 3.1.2.2 Unsupervised Learning

Unsupervised learning (UL) [101] is a type of machine learning in which the way a machine is trained includes uncategorized input data. The data does not teach the machine what output each input should have, thus avoiding problems of bias. Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. Rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output.

Unsupervised learning refers to the process of grouping data into clusters using automated methods or algorithms on data that has not been classified or categorized. In this situation, algorithms must learn the underlying relationships or features from the available data and group cases with similar features or characteristics. Therefore, this kind of training is much more similar to the process how human brain is working. Since childhood, humans are learning by relation and clustering elements, and sometimes it is even difficult for humans to distinguish between some images that actually are really different things. Humans don't need the solution (mean as label) of each object to know what is that thing, thanks to the association capability they usually have. It is so that unsupervised learning methods are so important for the development of machine learning, and specifically for some applications where there is not much data labelled.

The main tasks for unsupervised learning are feature extraction and characterization of separate and low-dimensional clusters. Unsupervised learning algorithms can automatically discover interesting and useful patterns in such data and they are suitable for creating the labels in the data that are subsequently used to implement supervised learning tasks. Besides, unsupervised clustering algorithms identify inherent groupings within the unlabelled data and subsequently assign label to each data value [4].

The figure 3.4 represents the diagram of the workflow of an unsupervised learning model represents the idea that has been expressed previously, where the input of a model is not labelled, and then the algorithms allow the clustering by extracting features. For this case, the accuracy of the model depends on the algorithm used.

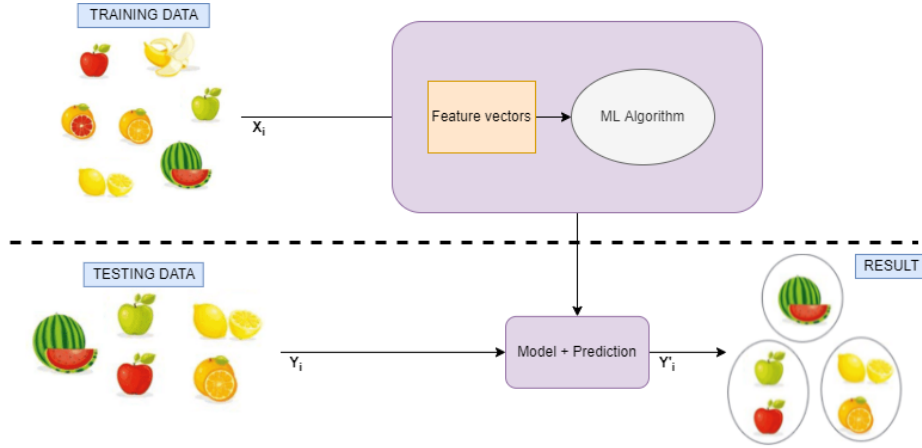


Figure 3.4: Unsupervised Learning Diagram

As it can be seen in the figure 3.4, the result for this prediction is the clustering of the different elements that have been used for testing the model. In this case, it can be defined which is the expected result, for example, apples, so the apples will be labelled with a positive result and the other clusters with negative result. The possibility to label a dataset by using unsupervised learning is very useful and enhance a lot the development of this type of Machine Learning.

In addition to that application, unsupervised learning models have been successfully used for other applications including pattern recognition, market basket analysis, web mining, social network analysis, information retrieval, recommender systems, market research, intrusion detection, and fraud detection [4].

Taking into account all the aspects, actually the two commented methods (Section 3.1.2.2 and Section 3.1.2.1) are totally different talking about the implementation of models and the applications they are used for. The Table 3.1 sums up the main differences between them.

Despite of the fact that the two types are totally different, they can be used together, creating then another type of ML called semi-supervised learning, which can make use of both labelled and unlabelled data and can be useful in application domains where unlabelled data is abundant, yet it is possible to obtain a small amount of labelled data.

Although this type of machine learning may seem to be the most convenient, in reality it is not always the most convenient. While it eliminates the problem of the volume of data required, on the other hand, it introduces the corresponding error of each type of learning. First, it introduces possible categorisation errors in the unsupervised model. Secondly, it may include problems of bias introduced by manual categorisation of the categorised dataset. This is why, as in the previous cases, it is designed for some specific applications

Properties	Supervised Learning	Unsupervised Learning
Input Data	Labelled $X_i, Y_i$	Unlabelled $X_i$
When to use	Known problems	Unknown problems
Applicable in	Classification and regression	Clustering and association
Algorithms (typically)	SVM, Decision Trees, Naïve Bayes	K-Means, PCA, Gaussian Models
Use cases	Spam filters, demand forecasting, price prediction, image recognition	Anomaly detection, customer segmentation, data categorization

Table 3.1: Comparison between SL and UL [7]

such as text classification or GPS way-finding.

### 3.1.2.3 Reinforcement Learning

Reinforcement learning (RL) is another ML paradigm that has received significant interest in the recent past. In contrast to other learning paradigms, reinforcement learning leverages a reward signal in order to learn a model, instead of using a supervisor. Thus, this reward signal can provide much higher level “labels” that the system can eventually use to learn from and can work particularly well in complex environments where it may be difficult to tease out specific rules that need to be learned.

Nowadays, reinforcement learning is recognized to be less valuable for business applications than supervised learning, and even unsupervised learning. However, many experts recognize RL as a promising path towards Artificial General Intelligence (AGI), or true intelligence, and it is being successfully applied in areas where huge amounts of simulated data can be generated, like robotics and games. Research lines are currently based on seeking ways to make RL algorithms more sample-efficient and stable. Some of the most remarkable current topics in RL are:

- Multi-agent reinforcement learning, this approach has recently demonstrated how the agents in a simulated hide-and-seek environment were able to build strategies that researchers did not know their environment supported, or how multiple agents influence each other if provided with the corresponding motivation.
- Off-policy evaluation and off-policy learning, for batch policy learning under multiple constraints, combining parametric and non-parametric models, and introducing a

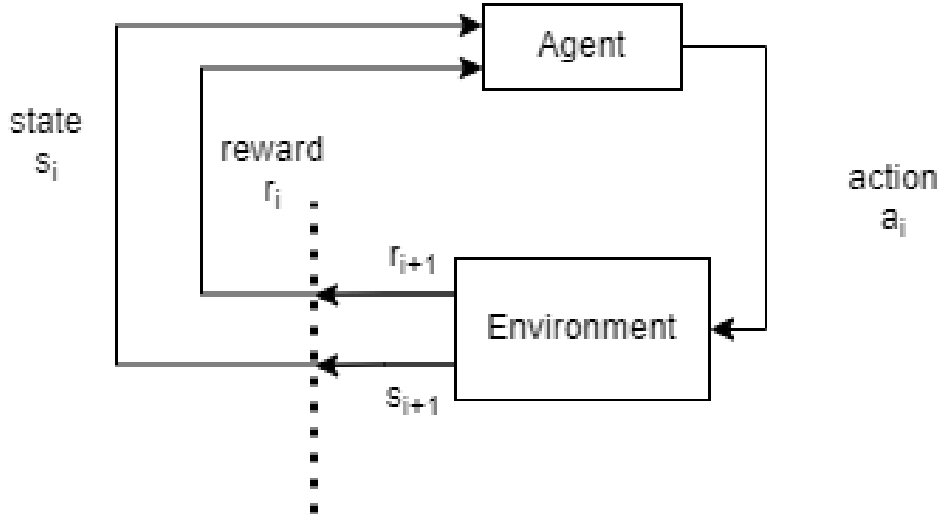


Figure 3.5: Reinforcement Learning diagram [32]

novel class of off-policy algorithms to force an agent towards acting close to on-policy. In off-policy reinforcement, learning algorithms are able to learn from data collected by a behavioural policy.

- Contextual bandits, also named associative reinforcement learning or one-step reinforcement learning. The algorithm observes a context, makes a decision, choosing one action from a number of alternative actions, and observes an outcome of that decision. The outcome defines a reward and the algorithm pursues to maximize the average reward.

Despite of not being the most used nowadays, there are still some important applications like Robot Navigation, Game Artificial Intelligence or Real-Time Decisions. These three applications could be joined in a real example like Tesla’s autopilot (driverless cars).

The Figure 3.5 represents the typical architecture of RL models [32]. RL can be formulated as a Markov decision process of an agent interacting with the environment to maximize the future reward. At each time step “ $i$ ”, given the current state “ $s_i$ ” and the current reward “ $r_i$ ”, the agent needs to learn a strategy (i.e., the “value function”) that selects the optimal decision or action “ $a_i$ ”. The action will have an impact on the environment that induces the next reward signal “ $r_{i+1}$ ” (which can be positive, negative, or zero) and also produces the next state “ $s_{i+1}$ ”. The RL continues with a trial-and-error process until it learns an optimal or sub-optimal strategy.

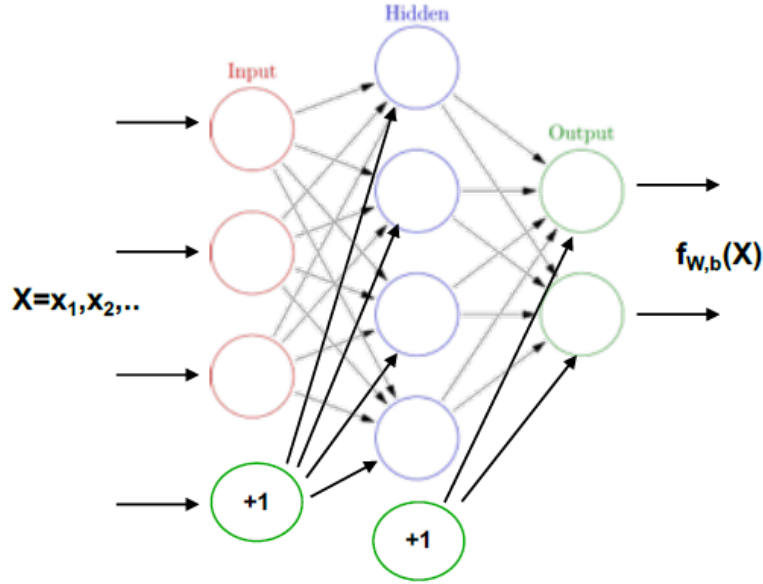


Figure 3.6: Deep Learning - Neural Network [31]

### 3.1.2.4 Deep Learning

Deep Learning [31] is a concept fully based on neural networks, which are a popular machine-learning technique that are largely used for supervised learning but can be applied to unsupervised learning problems as well. A neural network consists of inputs, layers, outputs, weights, and biases. The network shown in Figure 3.6 is an example of a feed-forward neural network that consists of one input layer, one hidden layer and one output layer. The arrows in the figure indicate connections across neurons.

In general, for a neural network of  $k$  layers, the following expression describes the input-output behaviour between any two layers ( $l$  and  $l+1$ ). Where  $y_l$  corresponds to the outputs of layer “ $l$ ”,  $W_l$  the weights between layers “ $l$ ” and “ $l+1$ ” and  $b_l$  the corresponding biases [31].

$$Y_l = f(W_{|Y|} + b_l) \quad (3.2)$$

During the training phase of a neural network algorithm, the weights  $W$  and biases  $b$  are iteratively adjusted in order to represent the often-non-linear relationship between inputs and outputs. The training phase consists of the following steps [31].

1. Forward Propagation: A training example (or set of examples) is fed through the network with some initialized weights and biases. The output is computed and an error term is calculated as the difference between the computed output(s) and the intended (ground-truth) output(s).

2. Backward Propagation: Given the error from the previous step, network weights are adjusted to better predict the labels of future unlabelled examples. This is done by adjusting each weight in proportion to its individual contribution to the overall error. As the contribution to the error from weights in earlier layers depends on the contribution from weights in later layers, the error signal is described as “backward propagating” through the network.

The above procedure is repeated for as many training examples as possible and when a particular convergence criterion (such as accuracy) is met, the model is considered trained and can be used for inference.

Currently the use of graph theory and the application of deep learning to graphs has shown very good results in the support to understanding the relationships between entities and concepts. Thereafter, techniques under Graph-based neural networks are capturing significant attention to provide enhanced performance in social networks and language processing as main examples.

The combination of reinforcement learning with deep learning has evolved into a field of research, called Deep Reinforcement Learning (DRL)<sup>2</sup>, which integrates the perception potential of deep learning and the decision making of Reinforcement Learning.

Deep neural networks (DNNs) [31] differ from traditional neural networks by having a large number of hidden layers. DNNs have had much success in the past decade in a variety of applications. Popular extensions of the neural network computation model include Convolutional Neural Networks<sup>3</sup>, Recursive Neural Networks<sup>4</sup>, and deep belief networks.

The use of the term deep indicates the difference between the use of a single or several hidden layers in the neural network. In that sense, the term reinforcement learning is related to a process of continuous learning through a trial-and-error method to optimize the outcome, while deep reinforcement learning refers to a learning process from previous knowledge that is applied to new data sets through deep learning technologies.

Therefore, Deep Reinforcement Learning can implement a variety of tasks requiring both rich perception of high dimensional raw inputs and policy control. As with other disciplines of machine learning and Deep Learning, the pace of development of Reinforcement Learning

---

<sup>2</sup>Deep Reinforcement Learning. Learning technique using only the input, reward and terminal signals together with the set of possible actions in a similar way as humans do.

<sup>3</sup>Convolutional Neural Network. Also named as Deep Convolutional Neural Networks or convnets, the architecture consists of three types of layers, namely convolutional, pooling, and fully connected. Oriented to process data that have a grid-like topology like images but as well to several others including text.

<sup>4</sup>Recursive Neural Networks. They can keep memory on past states of the network thanks to a feed-back capability. They provide memory to neurons allowing to capture information about what have been calculated so far.

and Deep Reinforcement Learning relies heavily on the increasing amounts of data, and the emergence of advanced Artificial Intelligence algorithms and powerful computer hardware.

Currently, Deep Reinforcement Learning faces the challenge of decreasing the time it takes to converge and learn something meaningful. This restricts the techniques from being used in the real world for real-time learning. This is why this field is having such a great success for example in video games (Atari) or strategy games whose rules are known (Go).

All variants of Reinforcement Learning involve trial-and-error learning: the agent takes random actions until it learns which behaviours earn the most total reward. This is generally problematic if learning is to take place in the real world, where errors can be expensive or deadly. Therefore, it seems that for practical applications Deep Reinforcement Learning techniques need to be either pre-trained, instead of datasets, in a simulated environments which reflect the real world, problem or application, to train and check the model, given that lots of iterations are needed before a Reinforcement Learning algorithm to work, or by transferring the learned knowledge between different domains.

Some of the most typical applications where Deep Learning [15] has been successfully used are biological analysis, image classification, Natural Language Processing, Automatic Text Generation and Drug Discovery.

### 3.1.3 Types of algorithm

Just as there are different types of machine learning, there are also different types of algorithms. Algorithms are the foundation of models, and depending on the type of application to be used, a different algorithm needs to be used. A first approach to divide the algorithms can be as follows. Some of the algorithms are further described in the following subsections<sup>5</sup>.

- Symbolic, logic-based and knowledge-based algorithms: Approach based in expression of concepts and processes by sets of symbols according to a limited set of logically defined rules and the use of background knowledge and constraints to define the search space. By the combination of different symbols following certain rules, complex ideas are formed and reasoning is possible.
  - Logic Programming. Learning hypothesis comprising a set of rules, given background knowledge and constraints for the search space and based on formal logic.
  - Expert Systems. These systems are based on knowledge and expertise providing reasoning capabilities for a specific area represented by ontologies, rules and

---

<sup>5</sup>The descriptions of the terms are to clarify the different concepts involved and should not be understood as definitions

information databases. The core components of expert systems are the knowledge base and the reasoning engine. They were the first operative form of AI algorithms

- **Artificial Neural Networks.** ANN are interconnected assemblies of simple processing elements, units or nodes whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.
  - **Deep Neural Networks.** Multi-layer neural networks addressing complicated notions by breaking it down into simplified characteristics through multiple layers. A further explanation is given in the Section 3.1.2.4.
  - **Graph-Based Neural Networks.** Neural networks that operate directly on arbitrarily structured graphs and which do not have an underlying Euclidean or grid-like structure, represented by graphs and manifolds or surfaces and not showing typical properties as global parameterization, common system of coordinates, vector space structure or shift-invariance.
- **Trees algorithms:**
  - **Decision Trees:** Non-parametric supervised learning methods used for classification or regression suffering normally from bias (for simple trees) and variance (for complex trees). Use of graphical representations of the possible options that can be selected in the resolution of a problem or a case, where internal nodes represent actions, arcs represent outcomes of an action, and leaves represent final outcomes. Construction of an optimal decision tree is an NP-complete problem and metaheuristics are normally used to solve them. This algorithm is better explained in the Section 3.1.3.1.
  - **Random Forests:** Machine learning algorithm that operates by constructing a multitude of decision trees at training time and providing as output the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- **Probabilistic methods:** Methods based on statistics and randomness rather than deterministic analytics.
  - **Logistic Regression.** It analyses data to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Estimates the probability of an occurrence of an

event based on one or more inputs. This algorithm is better explained in the Section 3.1.3.2.

- K-nearest neighbours (KNN). Simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. K-nearest neighbours captures the idea of similarity (sometimes called distance, proximity, or closeness) calculating the distance between points on a graph. This algorithm is better explained in the Section 3.1.3.3.
- Support Vector Machine (SVM). Support vectors are the data points that lie closest to the decision surface (or hyperplane). Support Vector Machine maximizes the margin around the separating hyperplane. This algorithm is better explained in the Section 3.1.3.4.
- Bayesian Networks. Also called Belief Networks. It describes systems by a directed acyclic graph, specifying relationships (arcs) of conditional dependence between its variables (nodes). A model is created by the graph, together with a joint probability distribution for the variables that can be used to make the inferences. Probability estimates are encoded for a large number of different competing hypotheses, with respective belief probabilities updated as new information becomes available. This algorithm is better explained in the Section 3.1.3.5.

### 3.1.3.1 Decision Trees

It should be remembered that supervised learning can be based on both classification and regression algorithms, and that decision trees can be used for both tasks. However, they are typically used more for classification. Similar to real trees, these algorithms have branches and nodes. The nodes represent the groups of variables to be classified, while the branches represent the values that the attributes can take.

Decision Tree [5] achieves classification in three distinct steps. Firstly, the algorithm induces both tree growing and tree pruning functionalities. Secondly, it grows the tree by assigning each data value to a class based on the value of the target variable that is the most common one at the instance of iteration. The final step deals with pruning the grown tree to optimize the performance of the resultant model.

A very typical example to explain this algorithm is the prediction of whether a tennis match will be played depending on some historic weather data. The data collected indicates if the game was or not played depending on some categories. The categories and the possible values are “Outlook”, with the possible values “Sunny, Overcast, and Rain”, “Humidity”, with the possible values “High and Normal”, “Wind”, with the possible values “Strong and Weak”, and “Temperature” with the possible values “Hot, Mild, and Cold”. As it can be

seen, the category “temperature” is not used, what means that the information that this category provides is useless.

The order of the categories in the tree is very important because is the one where the accuracy is greater. To compute the accuracy, the information gain is calculated based on the probability of each node. The probability of each node is calculated taking into account the historical data.

Also, it is defined the max depth of the tree as the number of levels it has. In the example above, the tree has 3 levels so the max depth is 3. By this way it can be predefined the number of splitting that the tree will have. This is a parameter that can be pre-configured in the models.

Some applications where Decision Trees have been successfully used are Medical Imaging (cancer detection), Star/Cosmic-Ray Classification in Hubble Space Telescope Images, Predicting Library Book Use, etc.

### 3.1.3.2 Logistic Regression

Logistic regression [13], despite its name, is a classification model rather than regression model. It is mainly used, as a Supervised Learning algorithm, for binary classification problems (where target is categorical data) but it can be also generalized to multiclass classification, which is called “Multinomial logistic regression”. This kind of algorithm achieves very good performance with linearly separable classes.

Logistic regression should not be confused with linear regression. The main difference between them is that logistic regression’s range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio.

Logistic regression essentially uses a logistic function defined in the equation 3.3 to model a binary output variable. In the logistic function, the “ $x$ ” is the input variable and the possible output values are always between 0 and 1 as commented before.

$$\text{LogisticFunction} = f(x) = \frac{1}{1 + \exp -x} \quad (3.3)$$

In the problems where Logistic regression is applied, the predictions will be classified as a class 0 if the probability (function result) is greater than 0.5, and as a class 1 in the other cases. The possible values provided by the function can be extended from minus infinity to positive infinity by using the logit function. The logit function is expressed in the equation 3.4, and the application of the logit function to the logistic function, assuming that the  $\text{logit}(P) = mx + b$ , is expressed in the equation 3.5. It is needed to be highlighted

that the equation 3.3 is the particular case where  $m = 1$  and  $b = 0$ .

$$\text{Logit}(P) = \ln \frac{P}{1-P} = mx + b \quad (3.4)$$

$$\frac{P}{1-P} = \exp mx + b \implies \dots \implies P(x) = \frac{1}{1 + \exp -(mx + b)} \quad (3.5)$$

Logistic regression [96] is used in various fields like most medical fields, and social sciences. Some examples of use of cases are Trauma and Injury Severity Score, risk of developing a given disease prediction, probability of failure of a given process, system or product prediction, marketing applications like customer's propensity to purchase a product, in economics to predict the likelihood of a person ending up in the labour force, and an extension of logistic regression to sequential data, are used in natural language processing.

### 3.1.3.3 K-Nearest Neighbours

K-nearest neighbour [13], also known as KNN is one of the simplest forms of supervised ML algorithm that is used for both classification and regression problems. KNN is assumed to be a nonparametric algorithm which means no assumptions are made about the underlying data. The KNN algorithm works based on the basis of similar proximity using distance calculations.

It is predefined the number of nearest neighbours (known as "K") that are going to be chosen. Then, it is calculated the distance (Equation 3.6) from the instance that is wanted to be classified with all the samples, and it is obtained the category (in classification problems) or numeric value (in regression problems) of the "K" nearest neighbours. Finally, using average in regression or majority in classification, the instance value is predicted.

$$d_e(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.6)$$

Actually, the most difficult part of the algorithm is the election of the correct "K". Choosing a small K means a higher influence on the result. On the other hand, choosing a higher K could lead to a smoother decision boundary with lower variance but increased bias. For this issue, it is usually selected more than one K, to see which one would result in the highest accuracy, using a cross-validation technique.

The main reasons for using this algorithm are the ease of implementation, the possibility of making predictions without the need to train the algorithm beforehand, and the ease of understanding and configuring the parameters on which the algorithm depends. On the other hand, it also has some disadvantages especially in high dimensional datasets when applying the distance formulas and with the cost of making predictions, as it is necessary

to calculate and search for the distances to the closest samples among all the samples in the data. In addition, it also presents difficulties when calculating distances between categorical features.

K-Nearest Neighbours algorithm [41] has been successfully used in some applications like text mining, pattern recognition, data mining, intrusion detection, facial recognition and recommendation systems.

#### 3.1.3.4 Support Vector Machine (SVM)

All the applications of Support Vector Machine [4] are included toward classification and the tenet of the algorithm is computation of margins. As opposed to many of the ML algorithms where the objective function is to minimize a cost function, the objective in Support Vector Machine is to maximize the margin between support vectors through a separating hyperplane. The margins are defined as the distance between two supporting vectors separated by a hyperplane.

Support Vector Machine [13] draws margins as boundary between the classes in the provided dataset. Its principle is to create the margins such that the distance between each class and the nearest margin is maximized and in effect leading to the minimum possible classification error.

The implementation of the algorithm and its accuracy is dependent on its ability to margin violations and subsequent missclassification of classes on either side of the vectors. The algorithm assumes that the data are linearly separable so that the weight associated with support vectors can be drawn easily and the margin computed. For almost all the cases, the data is not linearly and perfectly separable. Therefore, the following two concepts are very important to consider in those cases.

- Soft margin: this parameter is known as “C” and represents the degree of tolerance that may be assumed by the Support Vector Machine algorithm. It is used to find a line for class separation but this line will tolerate one or few missclassified instances. Basically, this parameter is a trade-off between the margin wide and the capability to classify correctly training data. The larger the distance between the support vectors, the higher the chances that points are correctly classifier. Therefore, the more general is to use a lower “C”.
- Kernel tricks: when the data are not linearly separable, kernel tricks utilize existing features and applies some transformation functions to create new features. These functions can be specified as parameters in the algorithm. For example, “linear”, “poly”, “rbf (radial basis function)”, “sigmoid”, etc. The most popular, typically providing the highest accuracy, are “poly” and “rbf”.

In the “ideal” case, support vectors are equidistant from the hyperplane and help in building the Support Vector Machine. Support vectors are called so because if their position shifts, the hyperplane shifts as well. That means the hyperplane only depends on the position of support vectors.

Another example where it is used a different kernel trick, is dimensionality reduction, where the distribution of the samples obligates to use another kernel function. The idea in this case is to map the non-linearly separable data from a lower dimension into a higher dimensional space to find a hyperplane. The mapping function transforms the 2D non-linear input space into a 3D output space using the kernel function, and, therefore the complexity of finding the mapping function in Support Vector Machine reduces significantly by using Kernel Functions.

Support Vector Machine [102] can be used for both classification and regression problems. Support Vector Machines are helpful in text and hypertext categorization, as their application can significantly reduce the need for labelled training instances. Some methods for shallow semantic parsing are based on support vector machines. Classification of images can also be performed using Support Vector Machines. Hand-written characters can be recognized using Support Vector Machine. The Support Vector Machine algorithm has been widely applied in the biological and other sciences.

### 3.1.3.5 Naïve Bayes

Naïve Bayes [30] is just a restricted/constrained form of a general Bayesian network. It is enforced the restriction that a class node cannot contain parents, and that nodes corresponding to the attribute variables cannot contain edges between them. Bayesian Networks can be used for classification, but Naïve Bayes could outperform a general Bayesian Network in classification task, as it is mentioned in the chapter 3 of the paper. In fact, Naïve Bayes algorithm has gained its fame because of its background on Bayesian probability theorem.

Naïve Bayes is mostly considered a semi-supervised method, because it can be used either in clustering or classification tasks. When implemented as a technique for creating clusters, Naïve Bayes does not require specification of an outcome and it uses conditional probability to assign data values to classes and, as such, is a form of unsupervised learning. However, when used to classify data, Naïve Bayes requires both input and target variables and, as such, is a supervised learning technique. As a classifier, the algorithm creates Bayesian networks [4]. Naïve Bayes algorithm relies on Bayes’ theorem represented mathematical in the equation 3.7 to assign independent variables to classes based on probability [75].

$$P(H|D) = \frac{P(H) * P(D|H)}{P(D)} \quad (3.7)$$

In the equation 3.7, the “H” and “D” are events with defined outcome. The probability of “H” when the probability of “D” is known is defined in terms of the product probability of “H” and probability of “D” given the probability of “H” divided by the probability of “D”. The extension of the theorem in supervised learning is of the form represented in the equation 3.8, where the set of samples  $X$  represents the input attribute, for which conditional probabilities are computed based on the known probabilities of the target variables in the training dataset.

$$P(H|D) = P(x_i, \dots, x_n|H) = \prod_i P(x_i|H) \quad (3.8)$$

The most used applications for which this algorithm has been successfully used is data labelling for subsequent unsupervised learning verifications, real time prediction, text classification, spam filtering, binary classification and recommendation systems.

## 3.2 Natural Language Processing

Natural language processing is a field of machine learning that focuses mainly on the understanding, handling and generation of natural language by machines. This technology has been essential for the development of the thesis, as the analysis of texts written by human beings is the core around which the solution to the problem is identified and developed.

Applications include speech synthesis, language analysis, language understanding, speech recognition, speech synthesis, natural language generation, machine translation, automatic question answering, information retrieval and extraction. For this purpose, the algorithms described in section 3.1.3, among others, can be used. As well as any of the types of machine learning described in section 3.1.2.

Specifically for the development of this thesis, some Python libraries have been used to process, analyse and extract information about the samples, which in this case are English texts. The main advantage of the Python language, apart from its simplicity, is that most of the libraries used are public and open source. Some of the libraries used are presented below.

### 3.2.1 Pandas

Pandas [81] is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. It enables to create series and dataframes objects. It also enables all the functionalities (visualization, selection, indexing, setting and operations) to work with those objects. Dataframes are

essential for the development of the thesis, as all the data used to train the model, is in this format.

### 3.2.2 NLTK

NLTK (Natural Language Toolkit) [80] is an interface and a set of libraries for working in Python with Natural Language Processing techniques. Among others, it has more than 50 corpora to work with them, a lot of lexical resources and it is one of the best tools for working in the Natural Language Processing field. It provides a big variety of preprocessing and processing functions, of which the most important are classification, tokenization, stemming, tagging, parsing, and semantic reasoning. The following functions are the ones that have been used in the development of the thesis:

- **NLTK Corpora:** NLTK has a large collection of corpora. It contains many different corpuses that can be useful for processing texts since it contains different language dictionaries, classifiers by genre, dictionary of synonyms, etc.
- **Tokenization:** is the process of transforming a text into separated words or sentences called “tokens”. Firstly, the text is filtered for removing undesirable characters, expanding contractions, etc. Then, the tokenizer function splits the whole text by marking blank spaces or phrase endings as the reference point for this division. Finally, all the words are processed again for deleting possible mistakes in the transcription of the words.
- **PoS-Tagging:** is the process of assigning different tags related to Part of Speech<sup>6</sup> to each word in the text. For the development of the thesis, the process of PoS-Tagging has been carried out using the NLTK recommended PoS-Tagger, which is much enough for English texts.
- **Stemming and Lemmatization:** These processes analyse the meaning behind the words. It means that the processes reduce the conjugation of words to the common base or root. The difference between one and other is shown in the following tables.
- **Removing Stop Words:** Stop Words are those words that do not add meaning to the text. Therefore, it is useful to remove them. NLTK provides a corpus of Stop Words in English for knowing which are these words and facilitate its removing.

---

<sup>6</sup>Part of Speech (PoS): There are eight parts of speech in the English language: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection)

Stemming			Lemmatization		
Form	Suffix	Stem	Form	Morphological information	Lemma
Studies	-es	Studi	Studies	Third person, singular number, present tense of study	study
Studying	-ing	Study	Studying	Gerund of the verb study	study

Table 3.2: Stemming vs Lemmatization

- FreqDist: the component FreqDist of NLTK provides some functions for searching which are the most common words within a text, among others.

### 3.2.3 Stylomepy

Stylomepy [21] is an open-source library, programmed in Python, and developed by GSI – UPM. It is a library which allows to calculate different metrics about the style of a text. It is mainly used as the basis of a prediction model, based on the differentiation of the language between texts from different sources. For this differentiation, functions such as readability index, vocabulary richness, formality and coherence are used. The calculation of these metrics makes it possible to transform a natural language processing problem into a normal machine learning problem. Each text input will have assigned numeric values, as result of each metric, and these numerical values are used to predict the output.

### 3.2.4 Imblearn

Imblearn [42] is an open-source library that runs in Python. Imblearn is the library that has been used to solve the problem posed in section 3.1.1. This library provides different methods for balancing data sets. As mentioned above, there are several balancing methods, such as oversampling, where samples of the least dominant class are created or replicated, undersampling, where samples of the most dominant class are removed or reduced, and other types of balancing.

### 3.2.5 GSITK

GSITK (GSI Toolkit) [85] is a library on top of scikit-learn that eases the development process on NLP machine learning driven projects. It uses NumPy, pandas and related libraries to ease the development. The main features of the GSITK library are the Word2Vec

Features, that implements a generic word vector model, previously loaded a Word Embeddings model that has to be compatible with Gensim. It allows, among others, to transform a text into a numeric vector to work with it. GSITK includes the implementation of the SIMON feature extractor. However, to enhance performance, it is recommendable to use a more complete scikit-learn pipe that implements normalization and feature selection in conjunction with the SIMON feature extraction.

### 3.2.6 Gensim

Gensim [86] is an open-source Python library created by Radim Rehurek for unsupervised topic modelling and Natural Language Processing, using modern statistical machine learning. It includes some implementations of Word2Vec algorithms, Latent Semantic Analysis (LSA, LSI), Latent Dirichlet Allocation (LDA), TF-IDF and more. The most important point of this library in this project is that it allows to work with Word Embeddings models and provides a lot of functions to ease the work.

## 3.3 Evaluation Techniques

The classification model predicts the probability that an instance belongs to a class. Evaluating the model is a very important step to know how good it will work in real-world problem solving. In this section, some evaluation techniques will be presented. An evaluation technique is a metric that allows model comparison with objective values. Besides, other processing techniques around the results evaluation are presented.

First of all, it is needed to be introduced some concepts on which the following metrics are based. These metrics are the following:

- True Positive (TP) indicates the number of samples that, being originally positives, are classified as it. True positives are relevant when we want to know how many positives our model correctly predicts.
- False Positive (FP) indicates the number of samples that, being originally negatives, are classified as positive. False positives can lead to incorrect decision-making. In some applications this misclassification is critical, as in medical themes.
- True Negatives (TN) are the samples that the model correctly predict as negative. It also indicates a good classification, but in this case about the negative samples.
- False Negatives (FN) are the classification of positive samples as negative. In the same way as False Positives, it can lead to incorrect decision-making, which could be critical in some applications.

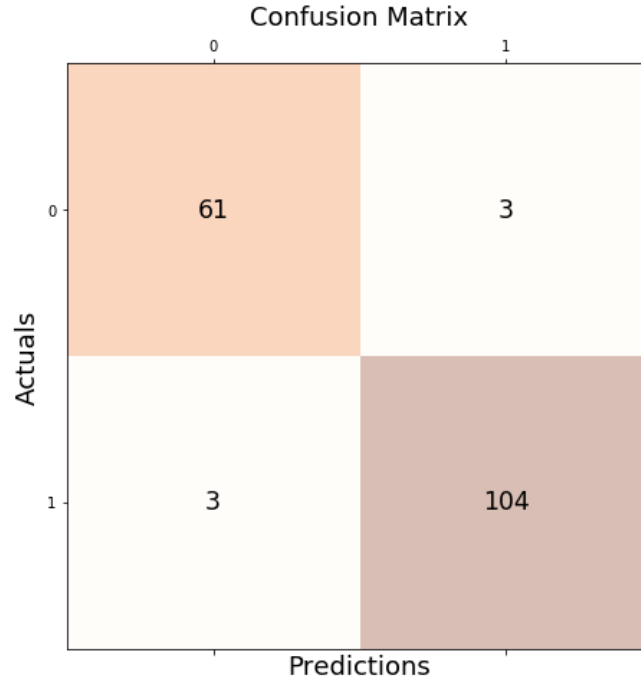


Figure 3.7: Evaluation Techniques confusion matrix [52]

These metrics are better understood with the confusion matrix. The confusion matrix represents how much data is classified in which form. It helps to understand some of the metrics that are computed in the following subsections. The figure 3.7 represents a confusion matrix, on which the true positive is 104 out of 107 (real positives), the false positive is 3 out of 64 (real negatives), the false negative is as well 3 out 107, and the true negatives are 61 out of 64.

### 3.3.1 Accuracy

Accuracy is defined as the ratio between true positive and true negatives to all the observations. This metric indicates how often the model will correctly predict a sample. Despite seems to be good, it actually doesn't indicate anything about the errors. The equation 3.9 represents the idea previously commented.

$$\text{Accuracy Score} = \frac{TP + TN}{TP + TN + FN + FP} \quad (3.9)$$

### 3.3.2 Precision

The precision score measures the proportion of positively predicted labels that are actually correct. Therefore, precision score is affected by the class distribution. If there are more

samples in the minority class, then precision will be lower. Precision is mainly used when we need to predict the positive class and there is a greater cost associated with false positives than with false negatives. The precision score is a useful measure of the success of prediction when the classes are very imbalanced. The equation 3.10 represents mathematically the concept.

$$\text{Precision Score} = \frac{TP}{FP + TP} \quad (3.10)$$

### 3.3.3 Recall

The model's recall score represents the model's ability to correctly predict the positives from among the true positives. It measures how good our machine learning model is at identifying all the real positives out of all the positives that exist in a dataset. Recall is also known as sensitivity or the true positive rate. Mathematically, it is expressed in the equation 3.11. Recall score can be used in the scenario where the labels are not equally divided among classes. For example, if there is an imbalanced dataset, then the recall score will be more useful than accuracy because it can provide information about how well the machine learning model identified rarer events.

$$\text{Recall Score} = \frac{TP}{FN + TP} \quad (3.11)$$

### 3.3.4 F1 Score

F1 score is a machine learning model performance metric that gives equal weight to both the Precision and Recall for measuring its performance in terms of accuracy. It's often used as a single value that provides high-level information about the model's output quality. Mathematically, it can be represented as a harmonic mean of precision and recall score, as it is shown in the equation 3.12.

$$\text{F1 Score} = \frac{2 * \text{Precision Score} * \text{Recall Score}}{\text{Precision Score} + \text{Recall Score}} \quad (3.12)$$

### 3.3.5 Cross Validation

Sometimes the data used to train a model is too concrete and exhaustive. The model will learn very well how to classify new inputs for a data type similar to the one it has been trained on. However, when you want to predict similar cases, but from other datasets, the machine learning model will not know how to compare the features extracted from the new data against the features it was trained on. In other words, the model cannot be used in

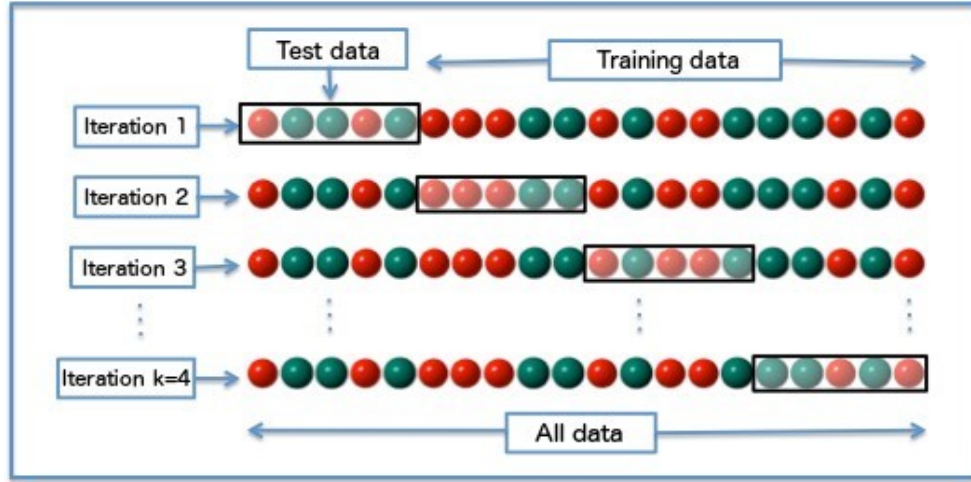


Figure 3.8: Evaluation Techniques - cross validation [76]

general, but only for the particular problem for which it has been trained. This fact renders the model useless, since it is useless to predict data for which its output is already known.

To face this problem, the method of cross-validation arises, which consists of a stage prior to training, in which a part of the data is separated and saved to test the model. A very useful process to perform this technique is K-Fold. This process consists of dividing the data into “K” groups and therefore “K” iterations. In each iteration, one of the groups is selected for testing, and the remaining “K-1” groups are selected for training. After training, the measures discussed in the previous sections are calculated, using the training and test data. This process is then performed “K” times, and finally, the arithmetic mean of all iterations is calculated. This idea is represented in the figure 3.8

### 3.3.6 Grid Search CV

Selecting the best hyperparameters when using different algorithms is not a trivial task. Each algorithm has its own hyperparameters, which can also take very different values. GridSearchCV offers the possibility to exhaustively consider all parameter combinations, and then, select the best one based on the accuracy. For this purpose, it is needed to specify the different values among the ones the method should try. Taking all the possible values, the GridSearchCV tests all the possible combinations.

## 3.4 Dashboard technologies

This section presents the dashboard that has been used to represent the results obtained in the project. Later in the section 4 it will be explained the process how this dashboard is created. This section only presents the enabling technologies for developing the graphical

interface. Data has been captured using **GSI Crawler**, then analysed and processed using the models developed for the thesis, then stored in **Elastic Search**, and finally shown in the **Dashboard**. All these tools are managed by **Luigi**, which is a task manager.

### 3.4.1 GSI Crawler

GSI Crawler is an innovative and useful framework that aims to extract information from web pages by enriching them following semantic approaches. It is possible to interact with the tool by a web interface, selecting not only the web page that is wanted to be examined, but also the analysis type that is wanted to be carried out.

GSI Crawler has been used for extracting web news related with terrorism. This news will be the basis of the final dashboard, as it is considered the only input to the pipeline. The sites that have been considered for extracting the news are the New York Times, the CNN, and Google News (all the public sources are included with this site).

### 3.4.2 Elasticsearch

Elasticsearch [17] is a search engine based on the Lucene library, developed by Spotify, which provides a distributed full text search engine with an HTTP web interface and a free JSON Schema document. Despite it has been developed in Java, it also has many other clients like C#, PHP, Apache Groovy, etc.

Elasticsearch takes in unstructured data from different locations, stores and indexes it according to user-specified mapping (which can also be derived automatically from data), and makes it searchable. Then Query DSL enable searching between indexed documents. Furthermore, the JSON interface makes even easier the use of this tool. Elasticsearch also provides the capability to scale from one to many instanced. However, this step requires a little bit more of expertise.

They key point that allows Elasticsearch to be so used, is the use of indexes. Indexes reduce complexity of data, and speed up data queries. Indexes are a kind of table where the results of the analysis carried out are stored. Each analysis will be stored in documents inside its corresponding index.

### 3.4.3 Graphical interface

The graphical interface has been developed with Javascript, and provides functions to represent, select, sort and filter the data that come from the Elasticsearch engine. The graphical interface is the last step of the pipeline. The data represented have been already analysed and processed. It is possible to select the indexes that want to be represented graphically.

The graphics are directly fed from the Elasticsearch engine, so it is possible to add any more data when is needed. This feature makes the Dashboard very versatile.

#### 3.4.4 Luigi

Workflow management systems are often necessary to manage the complex and demanding processes of Big Data environments. complex and demanding processes in Big Data environments. There are several open-source tools for workflow management, but in this case, it will be used Luigi, which was developed by Spotify.

Luigi [1] [19] provides a web interface to check pipeline dependencies, as well as an overview of the execution of tasks through an advanced user interface. It is published as a Python module, so it keeps coherence with NLP and ML tasks, as these ones have been also developed with the same programming language.

In this way, Luigi enables the use of pipelines, which are a series of interdependent tasks that are executed in order. Each task is defined by its input (its dependencies), its computation and its output. Luigi is used to orchestrate the different tasks between modules and ensures that there are no mistakes when sequencing tasks.

For this thesis, the data has been only extracted once, so the analysis and processing of it, it is only needed to be done one time. Therefore, Luigi indexes all data from Elasticsearch and Fuseki, and represents the graphics in the dashboard.

The other possibility was to implement a “real-time” <sup>7</sup> pipeline, where Luigi takes care of all tasks. First it retrieves the data via the GSI Crawler API. Then, in the processing and classification tasks, it analyses the data and makes the predictions. Finally, it indexes all the data in Elasticsearch and Fuseki, and represents the graphics in the dashboard. This option was not chosen because this is not a real production solution, and therefore, it is not needed the daily update of the data. It was considered better, the idea about extracting and representing the data on-demand. The hole pipeline is shown in the figure 4.1.

---

<sup>7</sup>It is established a periodically extraction of data from the predefined sources



## Architecture and Methodology

---

*This chapter presents the methodology used in this work. It describes the overall architecture of the project, with the connections between the different components involved on the development of the project. Starting with the collection of data, going through the pre-processing and application of the models and finishing with the visualization of the results. For the development of the project, some of the technologies that have been used are those commented in the Section 4.*

### 4.1 Introduction

One of the most important parts of a development project is precisely the representation of the results. There is no point in building a very powerful application code if the results are not interpretative afterwards. For this very reason, it has been decided to implement a dashboard where the obtained results are represented. To achieve this, a workflow has been implemented, which can be divided into several parts.

1. Obtaining articles from various sources would be the first of these, and where the process would begin. If there are no articles, there is nothing to represent. (Section 4.2)
2. Secondly, the extracted articles have to be processed. This processing includes the application of the model and the classification of the text as radical or not. (Section 4.3)

3. Thirdly, it is necessary to store the results in a database (Section 4.4, which will be the power source of the dashboard, where the stored results will be represented. (Section 4.5)

To build up the pipeline, Docker and Docker-Compose have been used. These are container-based technologies that allow the automation of the application deployment. The Dockerfile has made it possible to build the application environment, based on an existing python image. From this point, and by using Docker-compose, the rest of the services, the dependencies between containers and the networks have been defined. The figure 4.1 represents all the components used to build the pipeline, and the architecture followed. Among those services, the GSI Crawler, Elasticsearch, Redis, Kibana and Luigi, are each mounted in a specific container, and there are further explained in the following sections.

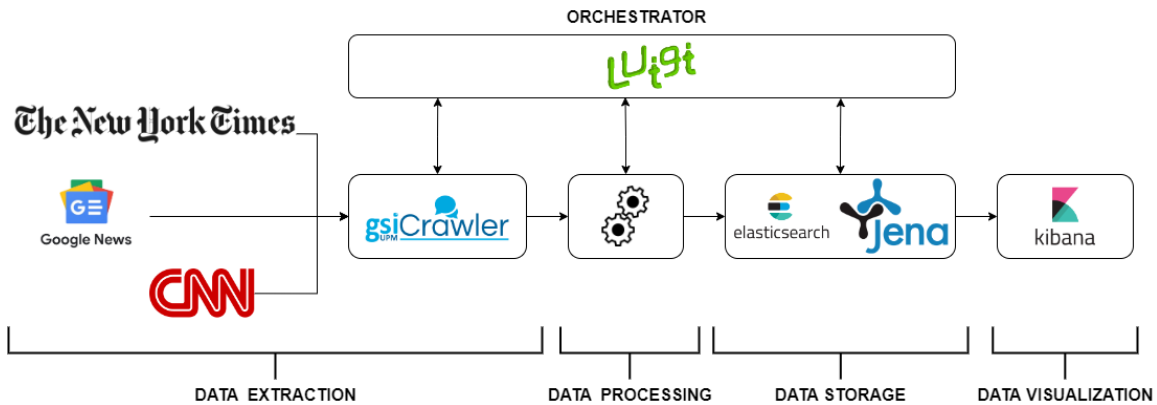


Figure 4.1: Pipeline architecture diagram

In order to coordinate and manage the services, Luigi has been used, which, as mentioned in section 3.4.4, is a service orchestrator based on concatenated tasks. The main advantage of this orchestrator is the possibility of visualizing the execution status of the tasks through a very intuitive panel.

Luigi allows the definition and execution of complex task dependency graphs and handles possible errors during execution. The task orchestrator allows you to stop, restart or skip the task manually. To do so, the container need to be running, and the user interface is accessible in localhost through the port 8082. In addition, it is also possible to see a lot of information about the process, like the graph of dependencies between the tasks, the workers, and the resources. Each task, which are defined by the input, the computation, and the output, communicates with the external service that performs actions on the data and transmits it through the pipeline. It ensures that there are no failures in sequencing the tasks.

The figure 4.2 represents the user interface of Luigi while a process is executed. In the

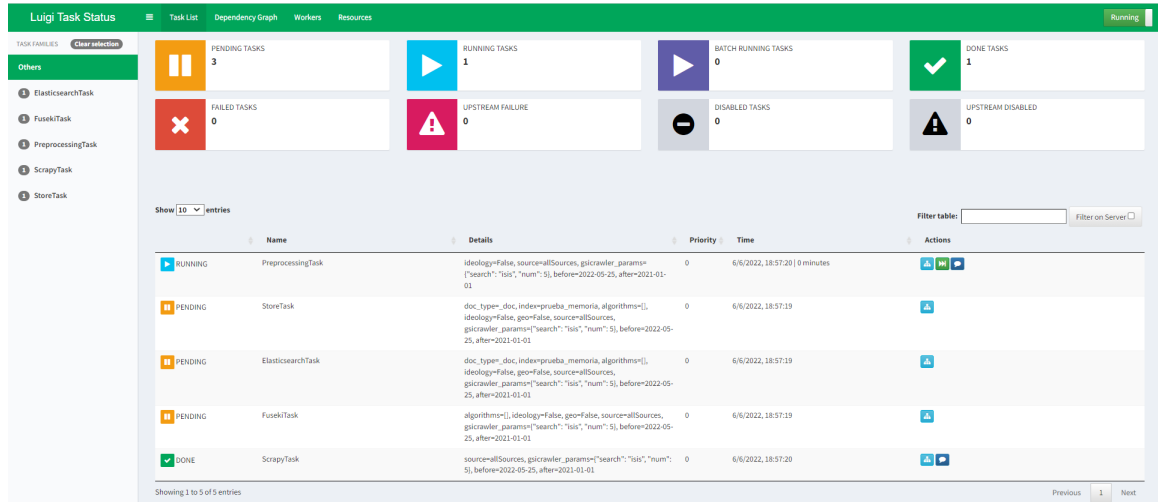


Figure 4.2: Luigi user interface

image, it can be seen that one task has been already done, while there is other running and other three pending. After finishing all the tasks, the results scraped and processed will be stored in the Elasticsearch database. Therefore, they can be represented in Kibana.

## 4.2 Data Extraction

This is the first step of the architecture, and it enables the whole pipeline to work. It mainly consists on the extraction of news from different internet sources. For this project, it has been defined three: The New York Times, CNN and Google News. For each of the sources, a customized scrapper has been developed. Scrapping is a method where a query is defined to search and extract some news from public/accessible sites. Many parameters can be set to extract the news. The most important ones are the content you want to search about and the dates you want to search between. For this thesis, the searched theme has been “isis”, and for the dates, it depends of the scraper used. For The NYT scraper and for the CNN scraper, it has been scraped news from January first of 2021 to June first of 2022. However, for Google News, as the API role used doesn’t have many permissions, only the last month news was extracted.

Ingestion is implemented by GSI Crawler container, the module in charge of retrieving the data. GSI Crawler functionality is based in Scrapy and other modules that connect to external APIs. This tool was developed by the GSI group of the university, and it is access-free. Table 4.1 represents the endpoints of the module.

This process has been executed only once, on June the third, and a total of 1690 results were scraped, as it is shown in the figure 4.3. It is clearly appreciated that the results

Endpoint	Description
GET tasks	Retrieves a list of available tasks in JSON-LD format
GET jobs	Retrieves a list of tasks. You can limit it to the pending jobs in execution by specifying “?pending=True”
POST jobs	Starts a new job, based on an available task and a set of parameters

Table 4.1: Possible endpoints for the GSI Crawler

extracted in the last month are much more than the ones before. This is due to the Google News API permissions.

It can be also appreciated what commented before about the last month news by looking into the figure 4.4, where it can be appreciated that from almost the whole period, the only news extracted has been those ones from CNN and The New York Times. As Google News is a very vast repository of news, the articles have been scraped from too many sources that the most of them are represented as “others”. However, it is still highlighted some newspapers like Al Jazeera English, Daily Mail or Freerepublic.

The output of this process is a JSON array with all the articles. Each entry of the array contains the main characteristics of an article, which are an ID, to identify the article, the source where the article comes from, the publication date, the headline of the article, the body, and the URL where it has been extracted from. This JSON will be the entry of the second task, the processing. The figure 4.5 represents an example about how the articles are scraped.

To implement this tool in a productive environment where it is defined a temporally slot to scrap news, it would be needed to have the containers running full-time. Unless this is something more than possible, it is out of the scope of the thesis. Therefore, the results shown were extracted at once. Actually, if the task is executed periodically, the representation of the data would be the same, but extending the horizontal axes to new dates.

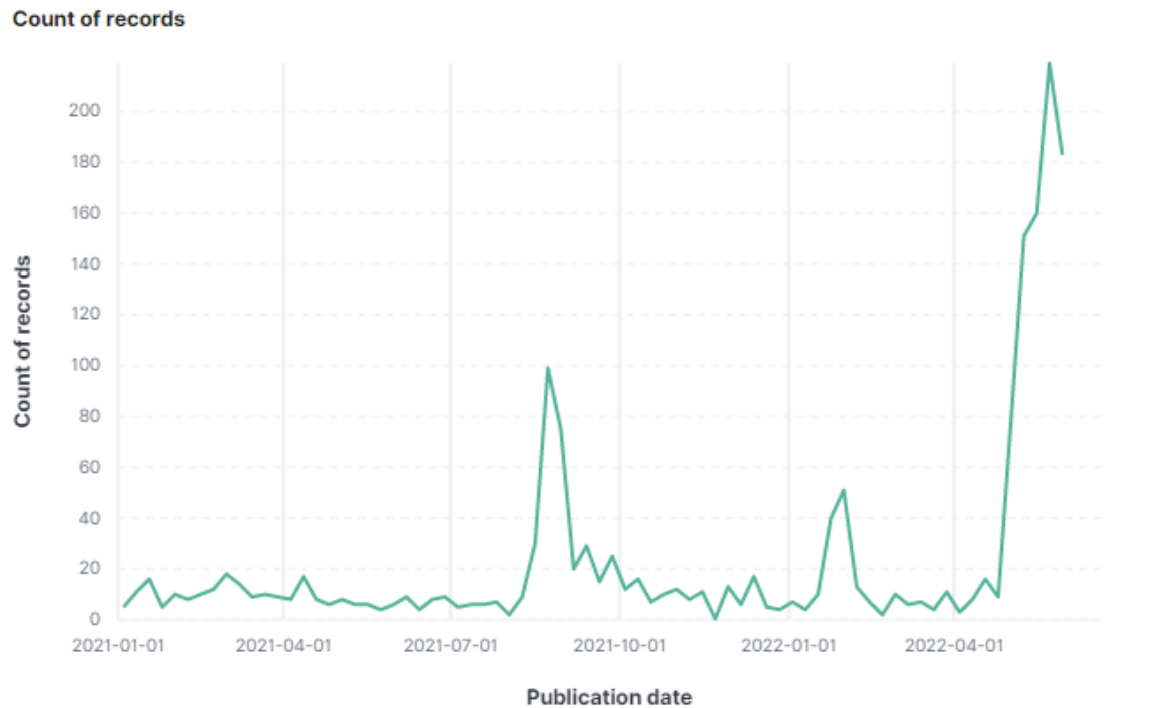


Figure 4.3: Count of articles scraped by date

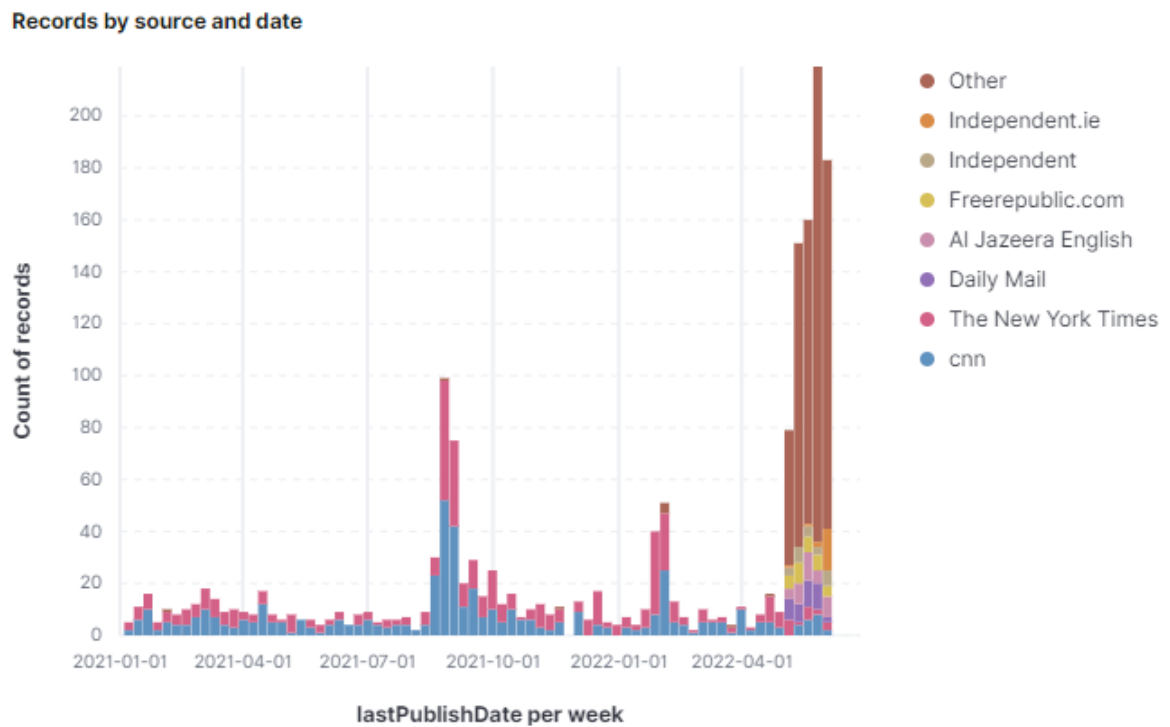


Figure 4.4: Count of articles scraped by date - grouped by source

```
{'id': '80024089',
 'source': 'Business Standard',
 'lastPublishDate': '2022-06-03T01:32:00Z',
 'headline': 'Rising attacks on people, places of worship in India, says Antony Blinken',
 'url': 'https://www.business-standard.com/article/current-affairs/rising-attacks-on-people-places-of-worship-in-india-says-antony-blinken-122060300084_1.html',
 'body': "US Secretary of State Tony Blinken said there has been rising attacks on people and places of worship in India, asserting that America will continue to stand up for around the world.\n\nHe also said people from the minority communities and women were being targeted in other Asian countries like Pakistan, Afghanistan and China.\n\nThe will continue to stand up for around the world. We'll keep working alongside other governments, multilateral organisations, civil society to do so, including next month at the United Kingdom's ministerial to advance religious freedom, Blinken told media persons at the release of the annual International report on Thursday.\n\nAlso Read: No locus standi: India rejects US global religious freedom report\n\nAt its core, our work is about ensuring that all people have the freedom to pursue the spiritual tradition that most has meaning to their time on earth, he said, noting that the report documents how religious freedom and the rights of religious minorities are under threat in communities around the world.\n\nFor example, in India, the world's largest democracy and home to a great diversity of faiths, we've seen rising attacks on people and places of worship; in Vietnam, where authorities harassed members of unregistered religious communities; in Nigeria, where several state governments are using anti-defamation and blasphemy laws to punish people for expressing their beliefs, Blinken said.\n\nChina, he said, continues to harass adherents of other religions that it deems out of line with the Chinese Communist Party doctrine, including by destroying Buddhist, Christian, Islamic and Taoist houses of worship and by erecting barriers to employment and housing for Christians, Muslims, Tibetan Buddhists and Falun Gong practitioners.\n\nIn Afghanistan, conditions for religious freedom have deteriorated dramatically under the Taliban, particularly as they crack down on the basic rights of women and girls to get an education, to work, to engage in society, often under the banner of religion, he said.\n\nMeanwhile, ISIS-K is conducting increasingly violent attacks against religious minorities, particularly Shia Hazaras, he added.\n\nIn Pakistan, at least 16 individuals accused of blasphemy were sentenced to death by Pakistani courts in 2021 though none of these sentences has yet to be carried out, Blinken said.\n\nBeyond these countries, the report documents how religious freedom and the rights of religious minorities are under threat in communities around the world, he said.\n\n(Only the headline and picture of this report may have been reworked by the Business Standard staff; the rest of the content is auto-generated from a syndicated feed.)"}
```

Figure 4.5: Example of retrieved articles features

## 4.3 Data Processing

The data processing task starts after data scraping has finished. As mentioned in the introduction, Luigi is the manager for deciding whether the first task has been completed successfully, so therefore, the second task can start. The processing task receives as an input a JSON file where the articles scraped are included. At this point, it is important to take into account that the scraper create each object of the JSON with predefined names. For example, the name of the object that contains the title is called “headline”. This must be taken into account for processing the data. Otherwise, if the method used in the processing task is changed, the pipeline could give an error.

The processing task consists basically on the creation of new features for each article. Three main features are added in this task. “Terms” and “headline\_terms”, which represents the words of the article and the headline respectively, after applying a tokenization method (later commented), and “Prediction”, which represents the prediction whether a text is radical or not.

On the one hand, to extract the “terms” features, a tokenizer method has been used. This method is the same one used during the development of the model for analysing the articles. The method consists mainly on applying regular expressions for cleaning the text, splitting the text into words, and extracting the lemma of each word. Finally, the set of words is returned.

On the other hand, for the “prediction” feature, it has been used a natural language processing model. In this case, due to the better results, it has been used the model

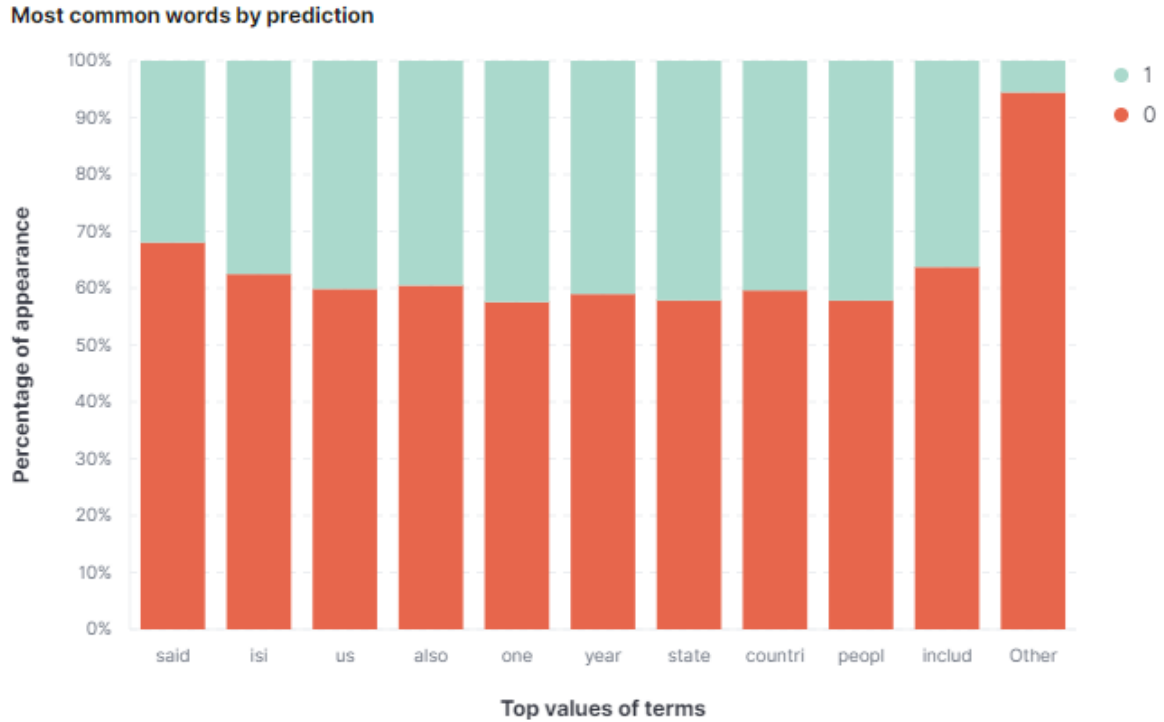


Figure 4.6: Most common words in articles grouped by prediction

presented in the section 5.2. Taking into account the body text, the text features are extracted and compared with predefined text features. Finally, it is returned “1” or “0” depending on whether the text is radical or not respectively. To see in more detail how the model is working, see the section 5.2.

These features have been represented at the final dashboard. The figure 4.6 represents the most common words in the articles text, and the figure 4.7 represents the most common words in the headlines. In both figures the terms are grouped by the prediction, and represented on a percentage graph. This graph shows the percentage of articles that have been predicted as “0 (non-radical)” or “1 (radical)” which contains the term.

As it can be appreciated in both figures, there are some common terms between body and title, like “us”, referring to United States, or “isi”, which is the lemma of “isis” and “isil”. Depending on the model used for predicting, the keywords could be very useful. In this case, the model that is applied is based on **similarity**, so individual words cannot be seen as clear evidence. Nevertheless, it is very striking that more than the half of articles containing the word “biden” in the title, are predicted as radical.

Also, it has been created a graph that remarks the difference between the predicted radical and non-radical articles. The figure 4.8 represents this idea. More than the 65% of the articles haven been predicted as non-radical.

And finally, the figure 4.9 represents the count of articles predicted by “1 (radical)” or

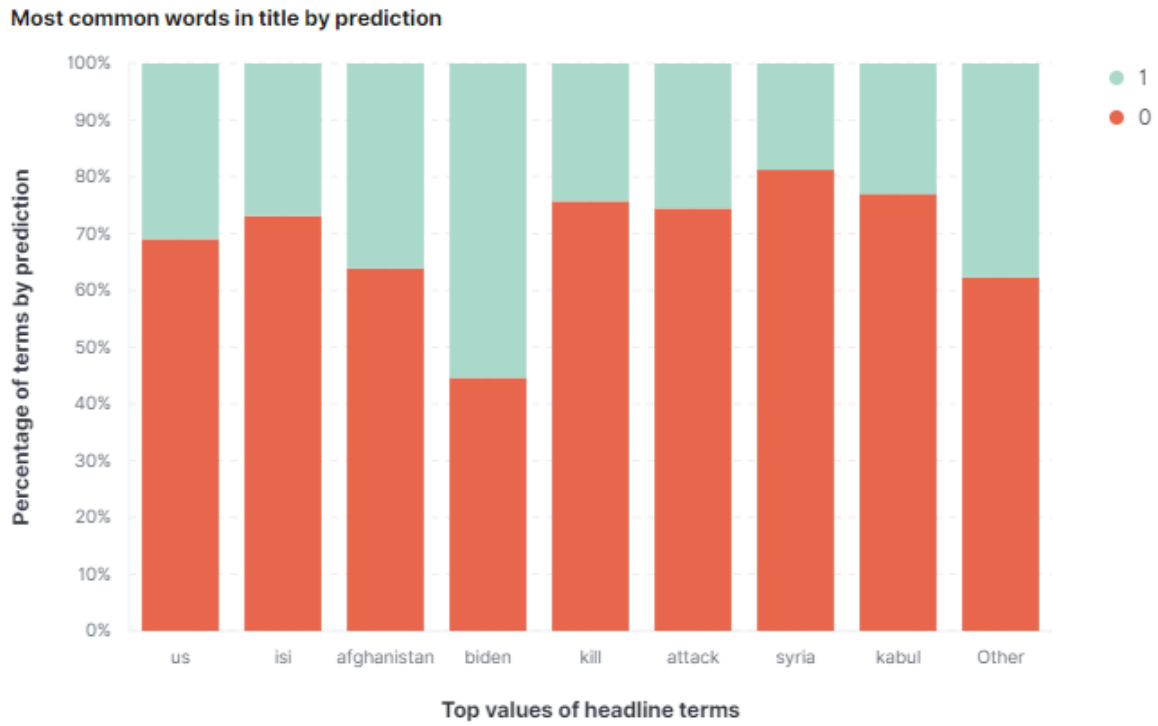


Figure 4.7: Most common words in headline grouped by prediction

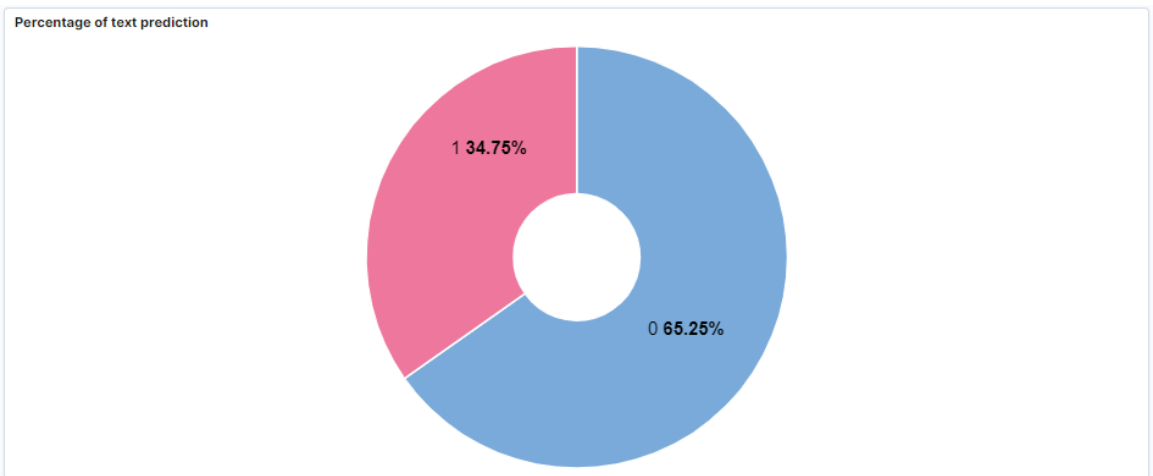


Figure 4.8: Total distribution of radical and non radical articles

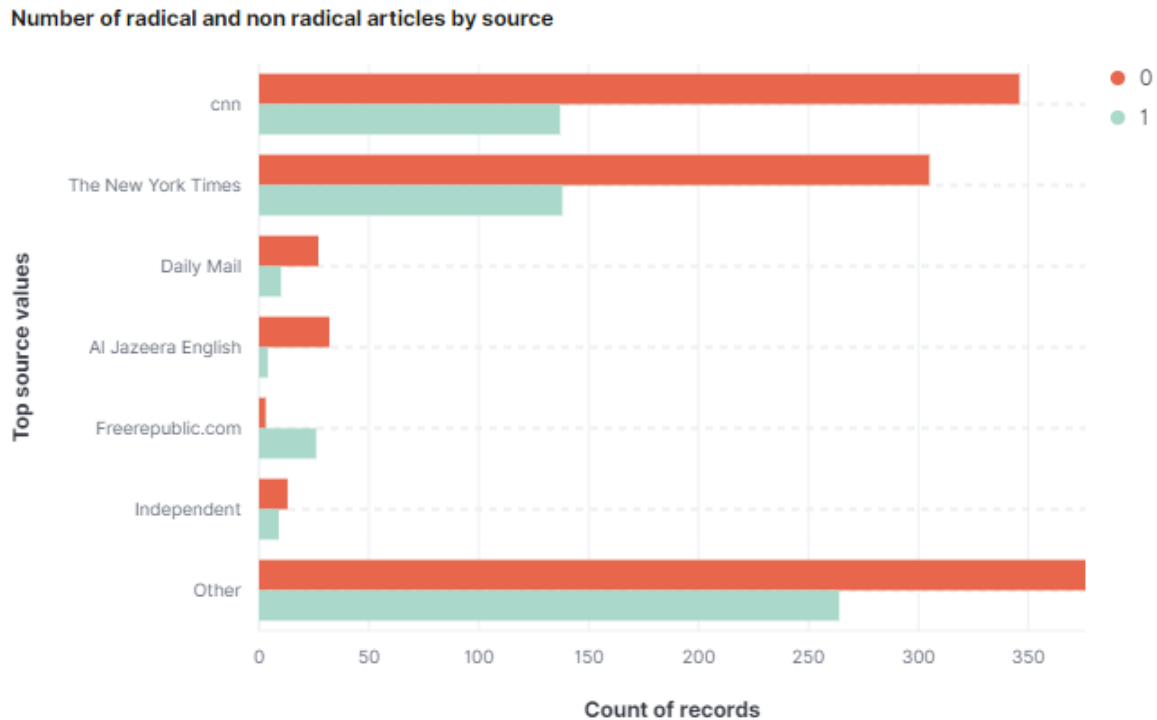


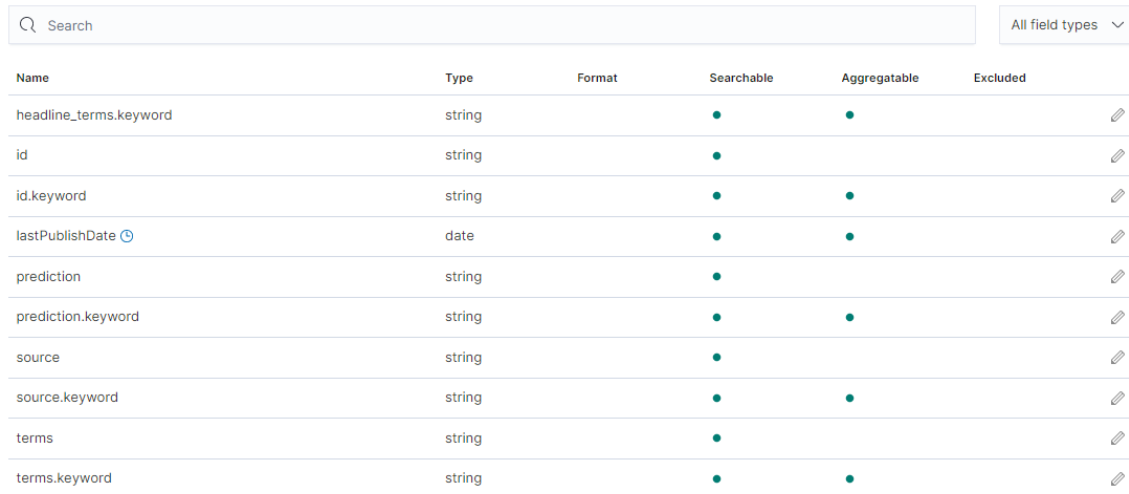
Figure 4.9: Number of radical and not radical articles by source

“0 (non-radical)” grouped by source. The results are astonishing for the source “Freerepublic.com”, where 26 out of 29 articles have been predicted as radical. This could lead us to the idea that this source has been publishing radical propaganda news during the month of May 2022.

As the output of this task, it is provided the same JSON got as input, but adding the commented features “terms”, “headline\_terms” and “prediction”. Actually, in the processing task, it could be done any operation needed to be represented. For example, other ideas could be to detect the language the article is written in. On the one hand, this could be great because articles from different languages can be scrapped. On the other hand, the model should be adapted and the difficulty would increase a lot, as depending on the language, texts are written differently. Other feature that could have also been added is the classification of the article from a political point of view (for example, “far left”, “left”, “center”, “right” and “far right”). However, this idea doesn’t provide advantages as the key point of the thesis is to predict radical articles.

## 4.4 Data Storage

After the processing stage, a JSON array with the features of each article is provided. The different JSON documents should be now stored. For executing this task, an ElasticSearch



Name	Type	Format	Searchable	Aggregatable	Excluded
headline_terms.keyword	string		•	•	
id	string		•		
id.keyword	string		•	•	
lastPublishDate	date		•	•	
prediction	string		•		
prediction.keyword	string		•	•	
source	string		•		
source.keyword	string		•	•	
terms	string		•		
terms.keyword	string		•	•	

Figure 4.10: Objects stored in ElasticSearch

database has been used. Each of the documents retrieved as an output of the processing task is appended to a specific index in ElasticSearch. This index is provided in the query that start the pipeline execution. At the moment when the pipeline is executed, a total of 1690 records were stored in the database. The storage task is the last one in the pipeline. However, this database is also used to feed the dashboard where the results are displayed. This kind of feeding allows the dashboard to be updated in real time whenever new data is coming to the database.

However, it is first needed to create an index pattern in the dashboard. When the Luigi pipeline finish, a new index with all the data is stored in the database, and this index is accessible and visible from Kibana, but being visible is not enough. To represent graphics in Kibana, an index pattern is needed. To create an index pattern, an index is needed. So, at the end, the index pattern needs to be created manually based on the index stored by the pipeline.

For improving the result, it is fairly recommended to have a date object in the documents. This object will allow ElasticSearch order the documents and therefore this order will be represented in the dashboard. For some of the string fields, others are automatically created by ElasticSearch, but they are actually the same, but with the possibility to be “aggregatable”. The figure 4.10 shows an example of some of the fields that are stored in an index of ElasticSearch.

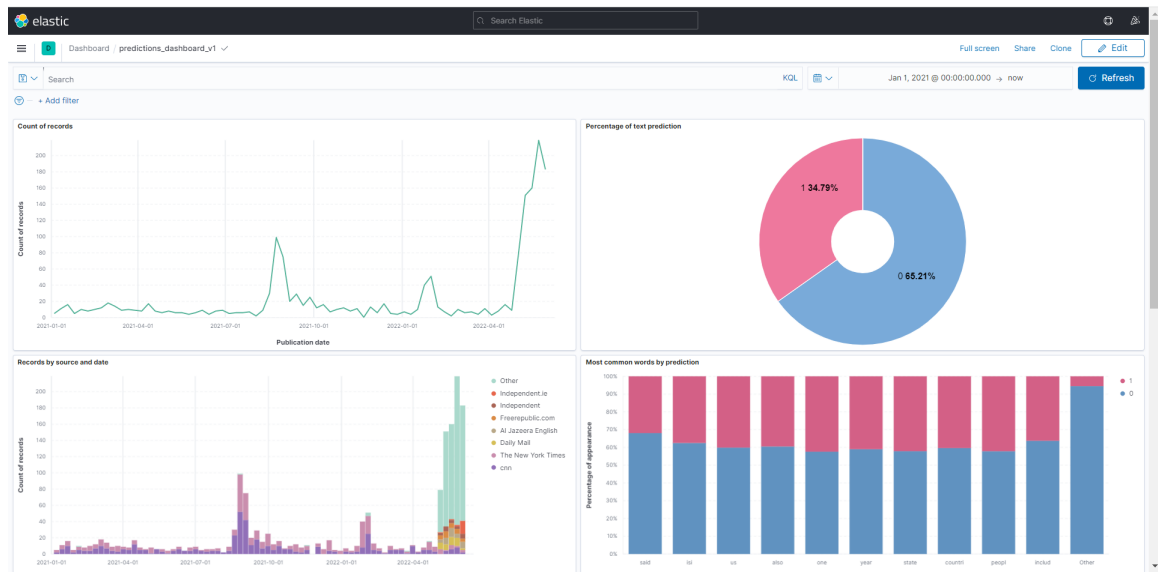


Figure 4.11: Dashboard Kibana

## 4.5 Data Visualization

After finishing the pipeline and when all the data is storage in the database, it is used Kibana as the visualization tool. Kibana is an open-source tool that allows data exploration. Many kinds of graphics are allowed. It is possible to combine more than one feature in order to see the relation between them. As it is directly fed from the Elasticsearch index, any change done in the database will be reflected in the visualization. However, it is first needed the creation of a dashboard. This dashboard will depend on an index pattern where the data is got from. In the dashboard, it can be configured the dates between you want to represent the data. Other options as filters are also allowed. The figure 4.11 represents the dashboard where all the graphs previously added are represented. Besides, it can be combined data from different index pattern in the same dashboard, as each graph depends on a specific one.



## Classification Model and Evaluation

---

*In this chapter, the models that have been developed are presented, describing the steps to build them and discussing the results obtained with each one. The first step that need to be done, is to import some libraries about Natural Language Processing, Machine Learning, Statistics, Graphic representation, etc. Between these libraries, there are all the ones commented in the section 3.2. Then, it is described the importation of data that is going to be used to implement the models. After importing the data, a pre-processing of it is made for each model. Then, some graphics are presented to understand and know the most important features. Finally, the classifier is tested and the results are presented.*

## 5.1 Data used

The development of this thesis has been done with jupyter notebooks and using Python. Therefore, to simplify the processing of the data, the importation has been done using dataframes. To do so, it has been imported the library “Pandas” (Section 3.2.1). By using this library, the data can be imported in different formats, but for this thesis, “json” and “tsv (tabulated separated values)” have been the ones used.

The datasets that have been used can be divided in two groups according to the features of each one. The first group consists only of one dataset, which is called “Radical Magazine Corpus”. This dataset is composed of articles extracted from four different public sources. Dabiq [92] and Rumiya [98] were online magazines used by the Islamic State of Iraq (ISIS) for the dissemination of propaganda and recruitment. Therefore, these magazines are considered radical propaganda. On the other hand, the CNN and New York Times, which are public newspapers of New York, have been the source from where the articles taking about ISIS, Daesh, Islamic State or others have been extracted. The following are the main features of this dataset:

- Id: identifier of the article in the dataset. It is used to make each line unique.
- Headline: title of the article in the magazine.
- ArticleBody: content of the article. This feature is the one that it is going to be processed by the model.
- Author: magazine where the article is published. At this moment, as this dataset has been created manually for the development of this thesis and others that belong also to the GSI group, it is the feature that make the label to be one or zero.
- label: decision about whether the article is considered radical propaganda (1) or not (0). As commented before, radical propaganda are the articles extracted from Dabiq and Rumiya, and non-radical propaganda are the article extracted from the CNN and the NYT.

On the other hand, the second group of datasets that have been used is called “Propy” [11]. The corpus was downloaded using MBFC metadata to identify propaganda vs non-propaganda sources. Specific URLs were then gathered with GDELT and contents downloaded with newspaper3k. For this group, there are three different datasets (dev, train and test) with the same features. Each of them can be used for a different purpose (development, training, and testing) and the main difference between them is the size of the data (5125 articles, 35986 articles and 10159 articles respectively). This data has been taken from different online

media such as US newspapers, online European articles, papers, etc. The main features for this group of datasets are the following:

- article text: content of the article that is going to be classified/predicted.
- event location: country or state where the article has been written.
- article data: date when the article was published.
- source name: name of the source where the article has been extracted of.
- MBFC bias label: political orientation of the source where the article was published. The main groups are extreme-right, least-biased, left, left-center, right, right-center, and unknown.
- propaganda label: categorization of each row, between radical propaganda (1) and not (-1). The labelling was done indirectly using a technique known as distant supervision, i.e., an article is considered propaganda if it comes from a news outlet that has been labelled as propaganda by human annotators.

The datasets have been imported into jupyter as dataframes. The figure 5.1 and the figure 5.2 represent the header and the tail of each dataframe in pandas, and include the main features that have been commented above.

	id	headline	articleBody	author	label
0	3d3b60a0	In the Words of the Enemy	Attacks continue\nof its operations to other r...	Dabiq	1
1	966f6826	THE ALLIES OF AL-QĀ'IDAH IN SHĀM: PART 4	that from the nullifiers of Islam was "backing...	Dabiq	1
2	ed36209b	Wisdom	A pagan church in Finland\nleft not knowing w...	Dabiq	1
3	c5d65817	TAWHĪD AND OUR DUTY TO OUR PARENTS	{And [mention], when Luqmān said to his ...	Dabiq	1
4	dd0802a1	THE VIRTUES OF RIBĀT FOR THE CAUSE OF ALLAH	al-Basrī \nsaid \n(rahimahullāh) \nAl-Hasan in...	Dabiq	1
...	...	...	...	...	...
2552	fd7fad4e	Trump administration unveils new counterterror...	President Donald Trump has signed off on a new...	CNN	0
2553	4231da90	Dutch police foil terrorist plot to target 'la...	Dutch police say they have foiled a major terr...	CNN	0
2554	c1f9432f	Justice for Victims of ISIS	To the Editor:\n\nRe "14 Death Sentences in 2 ...	The New York Times	0
2555	0ab1a3a6	Video Purports to Show Tajikistan Attackers PL...	A day after claiming its first attack in Tajik...	The New York Times	0
2556	c059c17f	A Second Chance for an Ivy League ISIS Recruit	"We've watched him closely," said Seth DuCharm...	The New York Times	0

Figure 5.1: Radical Magazine Dataframe

	article_text	event_location	article_data	source_name	MBFC_bias_label	propaganda_label
0	Convened to examine the causes of civil unrest...	Chicago, Illinois, United States	2018-02-27	The Hartford Courant	leftcenter06	-1
1	Discriminating against someone on the basis of...	Chicago, Illinois, United States	2018-02-27	The Hartford Courant	leftcenter06	-1
2	Bill Cosby's 44-year-old daughter, Ensa Cosby,...	Philadelphia, Pennsylvania, United States	2018-02-27	The Hartford Courant	leftcenter06	-1
3	The fast-moving, powerful theatrical locomotiv...	New York, United States	2018-02-27	The Hartford Courant	leftcenter06	-1
4	It's Friday. It's National Pizza Day. Grab lif...	West Hartford, Connecticut, United States	2018-02-09	The Hartford Courant	leftcenter06	-1
...	...	...	...	...	...	...
5120	<a href="https://www.lewrockwell.com/lrc-blog/carter-pa...">https://www.lewrockwell.com/lrc-blog/carter-pa...</a>	Moscow, Moskva, Russia	2018-10-22	lewrockwell.com	unknown	1
5121	Indeed, that title is worth repeating. The US ...	United States	2018-10-22	lewrockwell.com	unknown	1
5122	If the US ditches the Intermediate-Range Nucle...	Iran	2018-10-22	lewrockwell.com	unknown	1
5123	These opening comments will trigger knee-jerk ...	Washington, District of Columbia, United States	2018-10-22	lewrockwell.com	unknown	1
5124	"Here's where we hold 'em by the nose, and kic...	Italy	2018-10-23	lewrockwell.com	unknown	1

Figure 5.2: Proppy training Dataframe

As a common step for all the models, it is necessary to be checked how the data is balanced, since, as it was commented in the section 3.1.1, this is a fundamental aspect for training the model. To do this, simply distinguish the number of rows in each dataframe that belong to each category.

For the “Radical Magazines” dataframe, the data balance is shown in the upper Figure 5.3. The number of articles from each label is not balanced. There are a 67.5% of the articles tagged as non-radical (0) and a 32.5% tagged as radical (1). Therefore, the data needs to be balanced before training the models with it. For the other group of dataframes, “Proppy”, the data balance percentage is the same for the three of them. As commented before, the difference is the number of samples each one contains. It is shown in the second part of the Figure 5.3, the graphic which correspond to the “development” dataframe.

As it can be seen for all the dataframes, the data need to be balanced. Therefore, it is needed to know which of the balancing methods presented in the section 3.1.1 provides the best results. To do so, it has been tested the dataframes with a basic model and applying the different methods to compare the results and get the best one.

To test the results, the propy development dataframe and the “Radical Magazines” dataframe have been selected. First, a basic model based in Logistic Regression has been developed. The Propy development dataset has been used for training the model. The radical magazines dataset has been used for testing the model. The results obtained were very similar for all the balancing methods except of the method based on weights (explained in sec. 3.1.1.3) and the Threshold method (explained in sec. 3.1.1.4), for which the results were worse. Finally, the basic model was also tested with the Propy Test dataset.

According to the results presented in the table 5.1, the method that will be used to balance the training dataframe that will be used to train the models will be the Random

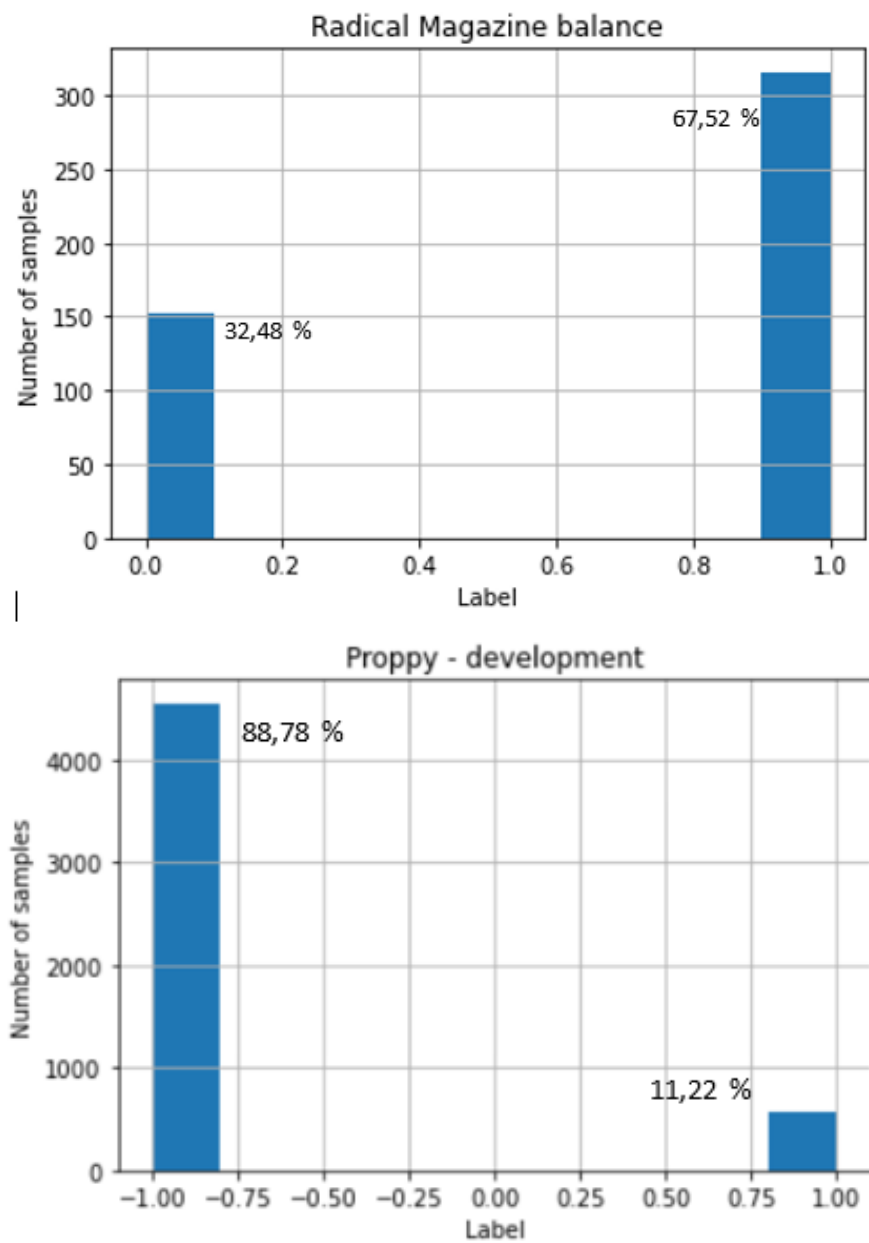


Figure 5.3: Dataframe label proportion before balancing data

Accuracy - Propy Development					
	Unbalanced	Random Over Sam- pler	SMOTE	<b>Random Under Sampler</b>	Near Miss
Propy Test	0.6775	0.78	0.79	<b>0.79</b>	0.7425
Radical Mag- azines	0.3632	0.6368	0.641	<b>0.641</b>	0.6645

Table 5.1: Balancing methods Accuracy

Under Sampler. This is because, although the results are the same as SMOTE, no “new” features are added to the model, over-training is avoided, and the execution burden is reduced by reducing the data load.

These dataframes are common for all the models that are presented in the following sections. For each model, a copy of the dataframes used have been created to keep the original one raw. The reason to create copies of the original ones is due to processing tasks. There are some rows of the dataframes that are not so useful. This could be because of many things like the article is too short or perhaps other features are empty. To detect this kind of things, the dataframes have been inspected to filter out the useless rows.

In addition to this, the “Propy” dataframe has two possible label values, which are “1” and “-1”. To be coherent with the radical magazines dataframe, the negatives values have been substituted by “0”. This change is not affecting the dataframe neither the results.

## 5.2 Model 1 - Similarity-based deep-learning model

Similarity is a natural language processing technique based on feature extraction to determine whether a text belongs with a higher accuracy to one or other class. The tag associated to each class could be any binary variable. For the development of this thesis, texts have been classified according to radicality (yes or no) from the point of view of detecting radical propaganda. In order for the model to exhibit this characteristic, it has been trained with the data presented in section 5.1.

### 5.2.1 Pre-processing of the data

Once the data has been extracted, it is necessary to perform a pre-processing of the data. This pre-processing is where the features that will train the model are extracted, so it is

a crucial step for the model. Furthermore, through pre-processing, the complexity of the data is reduced.

Transforming of the label column in Proppy dataframes and balancing the data are the first steps of pre-processing. These steps have been commented in the section 5.1, as these are general steps for all the models.

Then, the pre-processing of data is based on cleaning the articles by eliminating possible stop words, punctuations, URLs, etc., and applying lemmatization and stemming methods. To do so, apart from the lemmatization and stemming methods, regular expressions are used. This kind of processing can be done thanks to the NLTK library, as it was explained in the section 3.2.2, and thanks to the GSITK library, as it was explained in the section 3.2.5. The following is an example of a processed text.

Raw text: *“And [mention], when Luqmān said to his son while he was instructing him, ‘O my son, do not associate [anything] with Allah. Indeed, association [with him] is great injustice’...”*

Processed text: *“['mention', 'luqmn', 'said', 'son', 'instruct', 'son', 'associ', 'anyth', 'allah', 'inde', 'associ', 'great', 'injustic']”*

### 5.2.2 Data Analysis

This section presents an analysis of the data after pre-processing, in order to explain the distribution of the data, visualize the most important features and provide important information to train the classifier. It is important to check that the data is balanced before training the classifier. Machine learning classifiers strongly favour classes with a larger number of samples, classifying (almost) all samples as the majority class. In this case, as already presented in section 5.1, the best data balancing method for the propy training dataframe, is Random Under Sampler and it is the one that has been used to balance the data that will be used to train this model.

To compare the features, and to see which are the most relevant for the model, the distribution of some of the features is presented in figure 5.4. These categories belong to lexical analysis and PoS Tagging.

In addition, figure 5.5 shows the correlation between the features, including the label. Correlation measures the linear relationship that exists between two quantitative variables and that allows to explain and visualize the importance of each one. From this point of view, the most important thing will be the correlation between each feature and the label, as this implies the relevance that the model will assign to this feature in order to make the prediction. Between the features presented, it is determined that the most influential is the number of sentences.

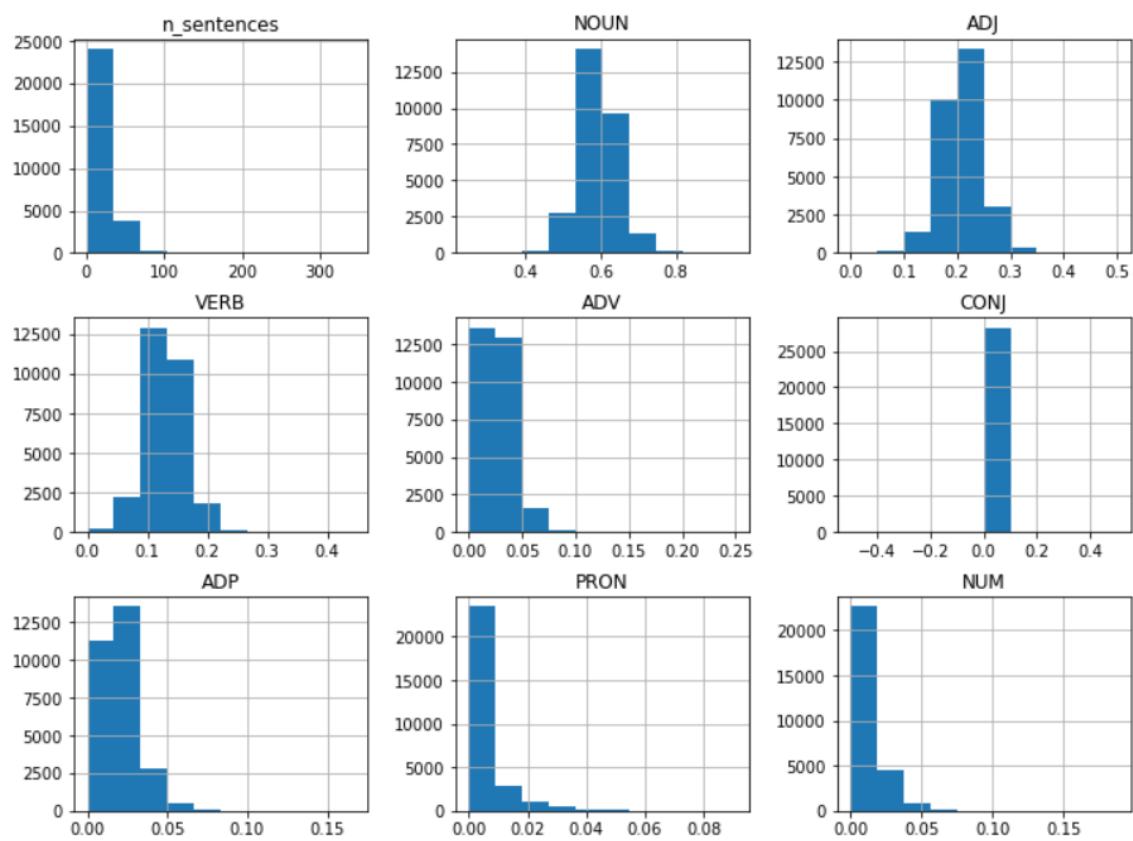


Figure 5.4: Histogram features Similarity-based DL model



Figure 5.5: Features correlation Similarity-based DL model

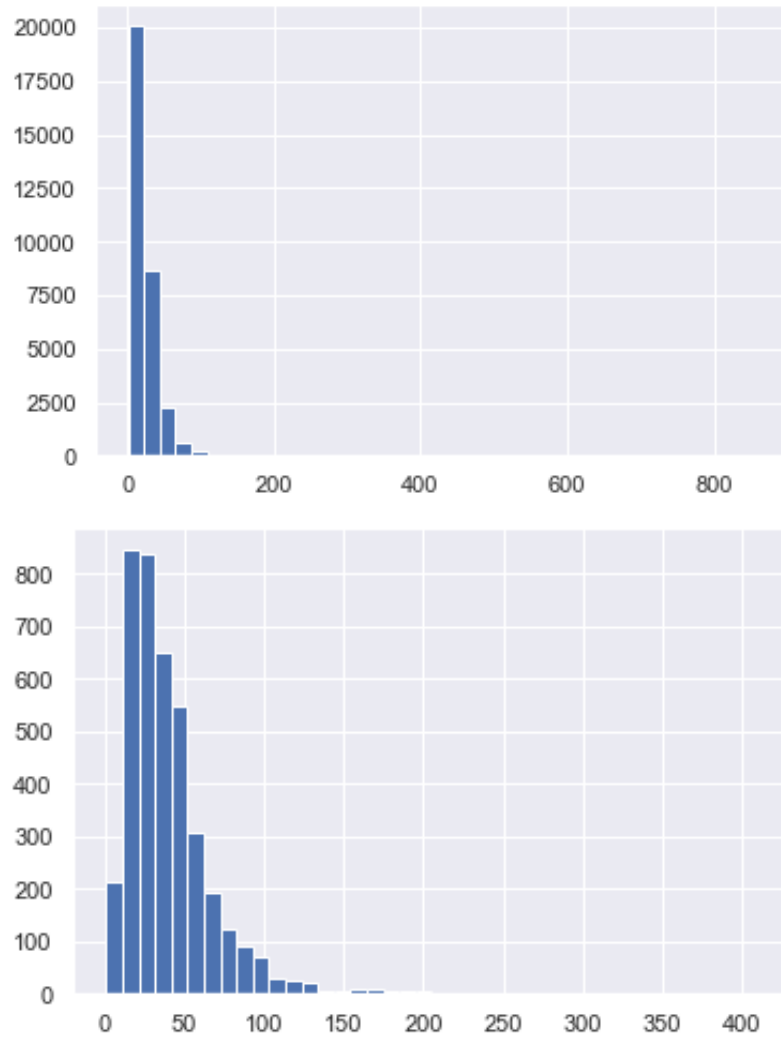


Figure 5.6: Histogram number of sentences - Similarity-based DL model

In fact, if the feature “n sentences” is extracted, it is possible to identify that the distribution between radical and non-radical texts is slightly different. The figure 5.6 represents this conclusion by splitting the graph into non-radical (left) and radical (right) and making a histogram of the results.

### 5.2.3 Feature extraction

This section presents the methods used for feature extraction in the Similarity-based deep-learning model. These features will be used to train the classifier and will therefore form the basis of the model. In this sense, what is important is not the quantity of features that are extracted, but the quality of the features. Many features can confuse the classifier. Some features lead to a more robust model, as they provide more valuable information.

To extract the features, there have been defined several methods. The process is a pipeline in which the output of one step is the input of the next one. The steps (methods) included in the pipeline are the calculation of the number of sentences each article has separated by an English tokenizer, sentences cleaning by eliminating possible stop words, punctuations, URLs, etc., application of stemming (English stemmer) and lemmatization (set of English stop words) methods, extraction of Part of Speech (PoS) analytics, application of SIMON model and similarity calculation. To calculate the similarity, it is needed an embedding model, for which is used a public one called “GoogleNews-vectors-negative300”.

As the final output of the pipeline, it is obtained a set of features extracted from the set of articles introduced in the pipeline and the label to which each article belongs to. In other words, the articles are divided by label, and then for each label, the main features of the articles are extracted and assign to this label. The output is a matrix with a dimension of (N×M), where N depends on the number of rows of the dataframe used, and M depends on the number of features that are extracted. In the case of the Propy training dataframe, N is 35.986, and M is 502.

The first set of features that have been extracted is named lexical stats, and consists on extracting the number of sentences each article has. Then, using a vectorizer, the list of feature-value mapping is transformed to a vector. The transformer that has been used is the “DictVectorizer” [72], which do a binary one-hot encoding (one boolean-valued feature is constructed for each of the possible string values that the feature can take on).

The second set of features that have been extracted is named Part of Speech (PoS) stats [97]. PoS is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. The words to be analysed are divided into eight groups, which are nouns, adjectives, verbs, adverbs, conjunctions, prepositions, pronouns and numbers. To do so, NLTK provides with a universal tagset that make the comparison between each word and the group it should belongs to. After labelling all the words, it is calculated the weight of each group by dividing the total elements of each category by the total elements of all the categories. Finally, the DictVectorizer is used to transform the dictionary list of features into a vector.

The third set of features that have been extracted is called word embeddings. Word embeddings is not a specific technique, but the name of a set of language modelling and feature learning techniques. Then, word embeddings can be used to represent words in text analysis, in the form of a real-valued vector that encodes the meaning of the word. In this way, the words that are near in the vector space, are expected to have a similar meaning. For this exercise, Word2Vec has been used. With this library, including a corpus text as input, will generate a vector space where each word will be represented with a vector.

For this project, it has been used the GSITK library with Word2Vec [8]. Word vectors

```
clf1 = linear_model.LogisticRegression()
clf2 = ensemble.RandomForestClassifier()
clf3 = tree.DecisionTreeClassifier()
clf4 = svm.SVC()
clf5 = naive_bayes.GaussianNB()
clf6 = naive_bayes.BernoulliNB()
clf7 = ensemble.GradientBoostingClassifier()
clf8 = ensemble.AdaBoostClassifier()
clf9 = neighbors.KNeighborsClassifier()
clf10=ensemble.ExtraTreesClassifier(n_estimators=1000, max_depth=None,
                                   min_samples_split=2, random_state=0, criterion='entropy')
```

Figure 5.7: Algorithms used in the models

are very computationally intensive. Moreover, the vectors themselves vary depending on the documents on which they are trained. Therefore, it is convenient to use word vectors that have been pre-trained, rather than training them from scratch for each project. Therefore, the model of Word Embeddings that have been used is given by Kaggle.

The last set of features that have been extracted is named Simon features. To do so, it has been used the GSITK library SIMON [8]. This technique is useful to obtain features **similarities**. With SIMON similarity computation, similarities are computed against a selection of lexicon words, and a max function is applied column-wise, obtaining a feature vector. The lexicon words are obtained by dividing the whole set of articles used in the model into radical and not radical. By this way, and applying SIMON features, the most meaningful words for each kind of article is extracted, creating the lexicon. It has been defined that the lexicon has a length of one hundred words.

### 5.2.4 Classification and evaluation

The last step of the model, is to make it able to classify new articles with the higher accuracy and f-score as possible. This step consists in training a classification model and evaluate it. For this, several algorithms can be used. First of all, it is needed to know which is the best algorithm for each model. In this case, the algorithms presented in the Figure 5.7 are going to be tested, which are mainly the one commented in the section 3.1.3.

The classification procedure is composed of two stages, training and testing. To do so, a dataset can be divided in two sets. However, for this thesis there will be used two different sets. As the first point needed is just to know which is the algorithm for which the best results are provided, the development propy dataset will be used. To test the model, the test propy dataset will be used. After training the model with the features extracted from the development dataset, the articles from the testing dataset are passed through the model, and the output is compared to the real output of the dataset. The main reason to use two dataset is to avoid dependencies, as commented in the section 3.3.5.

## 5.2. MODEL 1 - SIMILARITY-BASED DEEP-LEARNING MODEL

Algorithm used	Balanced data	Balanced data method	Accuracy	Precision	Recall	f1_score
GradientBoostingClassifier	True	Under sampling	0.839	0.839679	0.838	0.838839
RandomForestClassifier	True	Under sampling	0.818	0.815476	0.822	0.818725
ExtraTreesClassifier	True	Under sampling	0.819	0.823529	0.812	0.817724
AdaBoostClassifier	True	Under sampling	0.801	0.79802	0.806	0.80199
LogisticRegression	True	Under sampling	0.779	0.780684	0.776	0.778335
BernoulliNB	True	Under sampling	0.728	0.705776	0.782	0.741935
DecisionTreeClassifier	True	Under sampling	0.731	0.740125	0.712	0.72579
KNeighborsClassifier	True	Under sampling	0.636	0.614094	0.732	0.667883
SVC	True	Under sampling	0.657	0.650096	0.68	0.664712

Figure 5.8: Best classifier for Similarity-based DL model

Also in section 3.3, there were commented other evaluation metrics that are the ones that have been used to compare between different algorithms. Each of the algorithms has different parameters, called hyper-parameters, which can be tuned to obtain a higher accuracy or a higher f-score. These hyper-parameters allow to configure the algorithm and adapt it to the training data. Therefore, is not recommended to tune these hyper-parameters randomly. To help in finding which are the best hyper-parameters, GridSearchCV (section 3.3.6) can be used. This is a module of the library SKLEARN, and makes a search on the specified values of the parameters for an estimator. Depending on the hyper-parameter, it can be defined values by default or intervals of values and step of each interval to be tested. Then, all combinations are tested and the combination for which the success rate is maximum, is considered the best parameters for the algorithm.

The results obtained for each algorithm are shown in the Figure 5.8. The results are shorted by f1 score, and the best results are obtained for the Gradient Boosting Classifier algorithm. Specifically for this algorithm, the best hyper-parameters obtained were:

Best params: `{ 'clf_learning_rate': 0.5, 'clf_max_depth': 3, 'clf_n_estimators': 100 }`

Once it is known the algorithm and the balancing method that need to be used, it is the moment to implement the complete model, training it with the propy training dataframe, and testing it both with the Radical Magazines dataset and the propy testing dataframe. For this model, the results that have been obtained are presented in the table 5.2.

Both unbalanced and balanced methods have been calculated, to see the difference between them. First of all, it can be appreciated that the accuracy and the f-score on the propy test dataset are a little bit better without balancing than with balancing. This is

Balancing method	Without balancing		Balancing with Random Under Sampler	
Tested with	Proppy test	Radical Magazines	Proppy test	Radical Magazines
Accuracy	0.937	0.560	0.852	0.714
Precision	0.956	0.910	0.98	0.839
Recall	0.973	0.386	0.851	0.712
F-score	0.965	0.542	0.911	0.77

Table 5.2: Similarity-based DL model results

due to the model has been trained with the propy training dataset. Therefore, the model has learnt exactly the features extracted from this kind of dataset. On the other hand, it can be appreciated that the results of the radical magazines dataset are totally different. Balancing the dataset helps the model to learn a more general knowledge. Then, the results of the propy dataset are a little bit worst than without balancing, but the model will be more robust and comprehensive, which, at the end, is the purpose.

Sometimes, using more and more data is counterproductive for the models, because, as commented in section 3.1.1, the models only learn the features for the dataset that has been used to train. Therefore, it has been also tested the same model, but using the 67% of the data to train the model. The results are a little better when predicting samples of a different source, as the model has learnt in a more general sense. The results are shown in the table 5.3.

Comparing both tables, it is seen that the results obtained by using the 67% of the data, is a 1% higher than the one obtained by using all the data. Other tries have been realized using less data, and the results don't improve (significantly) the current ones. Therefore, it can be concluded that the model 1 predicts correctly a **general text** with an score of 78%.

Balancing method	Without balancing		Balancing with Random Under Sampler	
Tested with	Proppy test	Radical Magazines	Proppy test	Radical Magazines
Accuracy	0.935	0.577	0.847	0.724
Precision	0.955	0.940	0.978	0.845
Recall	0.973	0.399	0.847	0.725
F-score	0.964	0.560	0.907	0.780

Table 5.3: Similarity-based DL model results using 67% of training data

## 5.3 Model 2 - Stylometry

Stylometry is the application of the study of linguistic style. It is usually used to attribute authorship to anonymous or disputed documents, but it can be also used for other purposes. Between the main metrics that will be described in this section, there are highlighted the Readability Index, Vocabulary Richness, Formality and Coherence. At the same time, these metrics can be measured and interpreted in different ways. For this thesis, the library Stylomepy from the GSI will be used [21]. The application of this library in this thesis, will be used to compare between news, articles or statements, expressed by terrorist groups. This model is independent on the first one, and all the developments carried out, doesn't involve any of the ones carried out in the first model.

### 5.3.1 Preprocessing of data

Once the data has been extracted, it is necessary to perform a pre-processing of the data. This pre-processing is where the features that will train the model are extracted, so it is a crucial step for the model. Furthermore, through pre-processing, the complexity of the data is reduced.

Transforming of the label column in Proppy dataframes and balancing the data are the first steps of pre-processing. These steps have been commented in the section 5.1, as these are general steps for all the models.

Then, the pre-processing of data is based on punctuation removal by using regular expression, as well as for the first model. The difference is that for this model, stemming and lemmatization methods are not used. In addition, there are applied other conditions

	Preprocessed			Post processed		
Dataframe	Length	Radical	Non radical	Length	Radical	Non radical
Proppy train	35986	4021	31965	35902	4020	31882
Proppy dev	5125	575	4550	5113	575	4538
Proppy test	10159	1140	9019	10135	1140	8995
Radical Magazines	468	316	152	462	316	146

Table 5.4: Length for each dataframe in Stylometry model

that the articles need to match to enable the classifier. The articles with less than 19 characters are removed. After applying these techniques, the final dataframes length is presented in the table 5.4.

### 5.3.2 Metrics calculation

Once we have extracted and pre-processed the data, it is possible to measure the style of each article. The style will be defined by four metrics, Readability index, Vocabulary Richness, Formality Measure and Coherence Measure. Some of them have other submetrics. It is needed to be highlighted that in most cases, the style of a text depends drastically on the language it is written. For this thesis, the only texts that are going to be considered, are in English. To achieve the computing of all the metrics that are going to be presented, the Stylomepy library, which was presented in the section 3.2.3, from the GSI, is needed [21].

#### 5.3.2.1 Readability Index

Starting with Readability index, this is a style metric that measures how easy or difficult is to read a text. It must not be confused with the comprehension. That a text is difficult to read, doesn't mean that it is difficult to understand and viceversa. Different indexes are going to be used for this purpose, ARI (Automated Readability Index), Fog Count and

ARI Score		
Score	Age	Grade Level
0-5	5-11	Primary school
5-8	11-14	Middle School
8-12	14-18	High school
+12	+18	College/University

Table 5.5: ARI Scores related to US grade level

Flesch Readability. Each of them has their own scale of values. All of these indexes are based on the length of the sentences or the words of a text or even the number of syllables that a text has.

- Automated Readability Index (ARI)

ARI [89] depends on the characters per word and on the words per sentence. The mathematical formula to calculate this metric is expressed in the Equation 5.1, where (character/word) represents the average of character per word, and (words/sentence) represents the average of words per sentence. The output of the mentioned formula represents the US Grade Level needed to read fluently the text in subject. The possible levels are shown in the Table 5.5.

$$ARI = 4.71 * \frac{characters}{words} + 0.5 * \frac{words}{sentences} - 21.43 \quad (5.1)$$

- Fog Count Index

Fog Count [93] uses the concept of easy and hard words. The difference between them is the number of syllables. Hard words are considered to have more than 2 syllables, and easy words less than 2. The mathematical formula to calculate this metric is expressed in the Equation 5.2, where (words/sentences) represents the average of words per sentences, and (complex words/words) represents the average of hard words. The output of the mentioned formula represents the US Grade Level needed to read fluently the text in subject. The possible levels are shown in the Table 5.6.

$$FogCount = 0.4 * [\frac{words}{sentences} + 100 * \frac{complexwords}{words}] \quad (5.2)$$

Fog Index		
Score	Age	Grade Level
6-8	11-14	Middle school
9-12	14-18	High school
+12	+18	College/University

Table 5.6: Fog Index related to US grade level

- Flesch Readability Index

Flesch Readability Index is divided into two types. First of them is the Flesch Reading Ease Index, that provides as output the Grade Level needed to read fluently the text but using a scale from 0-100. The second one is the Flesch-Kincaid Grade Level Index provides directly the Grade Level needed. This second one is considered a redesign of the first one. The mathematical formulas to calculate theses metrics are expressed in the Equation 5.3 and in the Equation 5.4, where (word/sentence) is the average of words per sentences, and (syllables/word) is the average of syllables per word. Both outputs are presented in the Table 5.7.

$$Flesch\ Reading\ ease = 206.835 - 1.1015 * \frac{total\ words}{total\ sentences} - 84.6 * \frac{total\ syllables}{total\ words} \quad (5.3)$$

$$Flesch\ Kincaid\ grade\ level = 0.39 * \frac{total\ words}{total\ sentences} + 11.8 * \frac{total\ syllables}{total\ words} - 15.59 \quad (5.4)$$

### 5.3.2.2 Vocabulary Richness

The second set of metrics that have been calculated are the ones corresponding to the vocabulary richness, which includes the following submetrics.

- Type Token Ratio (TTR)

Type Token Ratio [21] is the basis of all the rest Vocabulary Richness metrics. It is defined as the relation between the set of words of a text, which will be called from now on like tokens, and the set of all the different words of a text, which will be called from now on like types.

Flesch Readability Index		
Flesch Reading Ease Score	Flesch-Kincaid Score	Grade Level
+90	0-5	Primary school
65-90	5-8	Middle School
50-65	8-12	High school
-50	+12	College/University

Table 5.7: Flesch Index related to US grade level

$$TTR = \frac{types}{tokens} \quad (5.5)$$

The form by definition indicates that the longer a text is, the more tokens will contain, and then, the lower TTR becomes. This is because the types and the tokens are not linearly related. Therefore, in large texts, the TTR could not represent the real vocabulary richness. The following metrics are based on the TTR and have been arranged to deal with the mentioned problem.

- Mean Segmental Type Token Ratio (MSTTR)

Mean Segmental Type Token Ratio [61] splits the whole text into  $n$  segments, calculate the TTR for each segment, and then, apply the equation 5.6 to calculate the mean. The main problem of this metric is the division of the text in segments, as the tokens could vary a lot between them.

$$MSTTR = \frac{\sum_{i=1}^n TTR(segment_n)}{n} \quad (5.6)$$

- Moving Average Type Token Ratio (MATTR)

Moving Average Type Token Ratio [20] provides a mean average of the TTR calculated so many times as the tokens the text has, minus the size of the window that has been defined. In this case, the size of the window will be 100, what means that the MATTR will calculate the TTR, starting from the beginning to the token number 100, then from the second token to the 101, and so on. The Equation 5.7 represents what this metric means, where  $TTR[token[i], token[i + 99]]$  is the TTR applied to the segment starting on the token “i” to the token in the position “i+99”, and  $length(tokens)$  is the number of tokens in the text.

$$MATTR = \frac{\sum_{i=1}^n TTR[token[i], token[i + 99]]}{length(token) - 99} \quad (5.7)$$

- Measure of Textual Lexical Diversity (MTLD)

Measure of Textual Lexical Diversity [21] also allows to measure the vocabulary richness. However, it takes into account not only the tokens, but also the segments of the text, considered as the number of segments of the text that have a TTR lower than a threshold. In this case, the threshold used has been **0.72**, which is the typical one. This metric needs to take into account the last segment of the text, which is called Partial Segment, whose expression is shown in the equation 5.8. By this way, the equation 5.9 represents this metric.

$$Partial\ segment = \frac{1 - TTR}{1 - limit} \quad (5.8)$$

$$MTLD = \frac{tokens}{Segments + Partial\ segment} \quad (5.9)$$

- Hypergeometric Distribution of the Diversity (HD-D)

Hypergeometric Distribution of the Diversity [21] is based on the hypergeometric distribution, analyses the probability that any type will appear in a sample of **42** random tokens. Then, the equation 5.10, represents this idea, where the  $P(X = type_i)$  is the probability given by the hypergeometric distribution for each type, and whose expression is shown in the equation 5.11.

$$HDD = \sum_{i=0}^n \frac{1}{42} * P(X = type_i) \quad (5.10)$$

$$P(X = x) = \frac{\binom{d}{x} \binom{N-d}{n-x}}{\binom{N}{n}} \quad (5.11)$$

### 5.3.2.3 Formality measure

The third set of metrics metrics that have been calculated are the ones corresponding to the Formality Measure, which is the degree of formality of a text. To measure the formality, is necessary to know how many words of each type are in the text. Therefore, this metrics are based in the POS Tagging. There are highlighted two kinds of metrics, Adjective Score and F-Score.

- Adjective score

Text Category	Total tokens	ADJ Tokens	ADJ Density
Conversation	2,368,324	82,599	3.49
Other spoken	2,382,061	111,126	4.67
Fiction	2,382,786	139,894	5.87
Newspapers	2,360,843	159,046	6.74
Unpublished writing	2,395,601	162,826	6.80
Other published writing	2,354,825	197,100	8.37
Non-academic prose	2,451,482	213,128	8.69
Academic prose	2,468,802	237,709	9.63

Table 5.8: Adjective scores - categories and typical values

Adjective score [26], as the name indicates, measures the density of adjectives in a text. Typically, the metric is expressed in percentage, so the equation 5.12 represents this metric. Furthermore, depending on the adjective density, there are several categorizations for texts. The table 5.8 represents some categories and the score for each one. To extract conclusions from this metric, it needs to be taken into account also the Readability index, which indicates how difficult is a text to be read. As more adjectives, typically is more difficult.

$$Adj.score = \frac{N \text{ Adjectives}}{tokens} * 100 \quad (5.12)$$

- F-Score

F-Score [25] [39] is more valuable than adjective score because it takes into account all POS tags, not only the adjectives. The way this metric is calculated is expressed in the Equation 5.13. The more F-Score, the higher degree of formality. In addition, an approach to categorize the formality is shown in the Figure 5.9.

$$F \text{ Score} = \frac{\frac{Nouns + Adj. + Prepos. + Determiners - Pronouns - Verbs - Adverbs - Interjec.}{2} * 100 + 100}{2} \quad (5.13)$$

At the beginning it has been tried to calculate these metrics on the Propy training dataframe, as it was commented in the section 5.1. However, due to the size of the dataframe

	<i>"explicit" categories</i>				<i>"deictic" categories</i>				Conj.	Formality
	Nouns	Articles	Prepos.	Adject.	Pron.	Verbs	Adverbs	Interj.		
Movies A	13.37	8.29	8.62	5.08	1.62	27.03	9.98	0.77	5.98	<b>48.0</b>
Theatre A	14.84	10.15	9.44	5.51	1.43	24.48	8.71	0.77	5.55	<b>52.3</b>
Theatre B	13.96	10.22	10.54	4.83	1.37	23.94	8.14	0.13	7.17	<b>53.0</b>
Novels A	16.72	13.79	14.04	5.58	8.50	20.05	6.45	0.13	6.42	<b>57.5</b>
Novels B & Sh.Stories	18.19	16.03	15.45	6.74	7.04	17.65	4.45	0.07	6.27	<b>63.6</b>
News-papers A	18.92	16.80	16.73	7.70	5.10	17.51	4.86	0.02	5.15	<b>66.3</b>
Essays B	18.95	16.91	17.15	8.09	5.75	12.90	4.15	0.03	6.95	<b>69.1</b>
Newspap. & Magaz. B	20.41	18.35	18.39	8.35	4.29	15.41	3.47	0.01	5.27	<b>71.2</b>
Technical& Scientif. B	18.63	17.99	20.17	7.56	4.27	12.73	4.12	0.00	6.00	<b>71.6</b>

Figure 5.9: F-Score typical values [39]

	count	ARI	FleschRI	FogCount	TTR	MSTTR	MATTR	MTLD	HDD	Form_score	Adj_score
0	1198	19	15.192952	9.977273	42.556770	67.430712	51.042899	67.340955	70.364329	75.147183	17.665615
1	1485	16	13.523043	7.578125	37.975543	70.129630	48.598972	76.146095	74.065319	64.232337	17.079890
2	669	13	8.930320	3.615385	39.941263	67.985891	42.602198	73.822164	70.473024	53.817915	16.725979
3	1515	16	13.372659	7.866667	31.116850	65.534439	42.282286	59.977560	71.178949	65.978050	15.363128
4	1120	17	13.536499	9.197674	40.017746	70.299383	49.259554	80.846151	74.913092	70.008873	13.973064

Figure 5.10: Metrics results Stylometry model

(35902 entries that match the requirements), it took almost two days to process all entries. Therefore, the radical magazines dataframe has been used for this purpose, which takes 56 minutes in extracting all the metrics.

Applying this metrics to the data, the results presented (only for the first five rows) in the figure 5.10 have been obtained. It is needed to be highlighted that the output of the method that calculates all the metrics, is a new dataframe which will be used to train the model. Each column of the dataframe is a feature from which the model will learn. Not all the features have the same relevance, but this aspect will be commented in the next section.

### 5.3.3 Data Analysis

This section presents an analysis of the data after pre-processing, in order to explain the distribution of the data, visualize the most important features and provide important in-

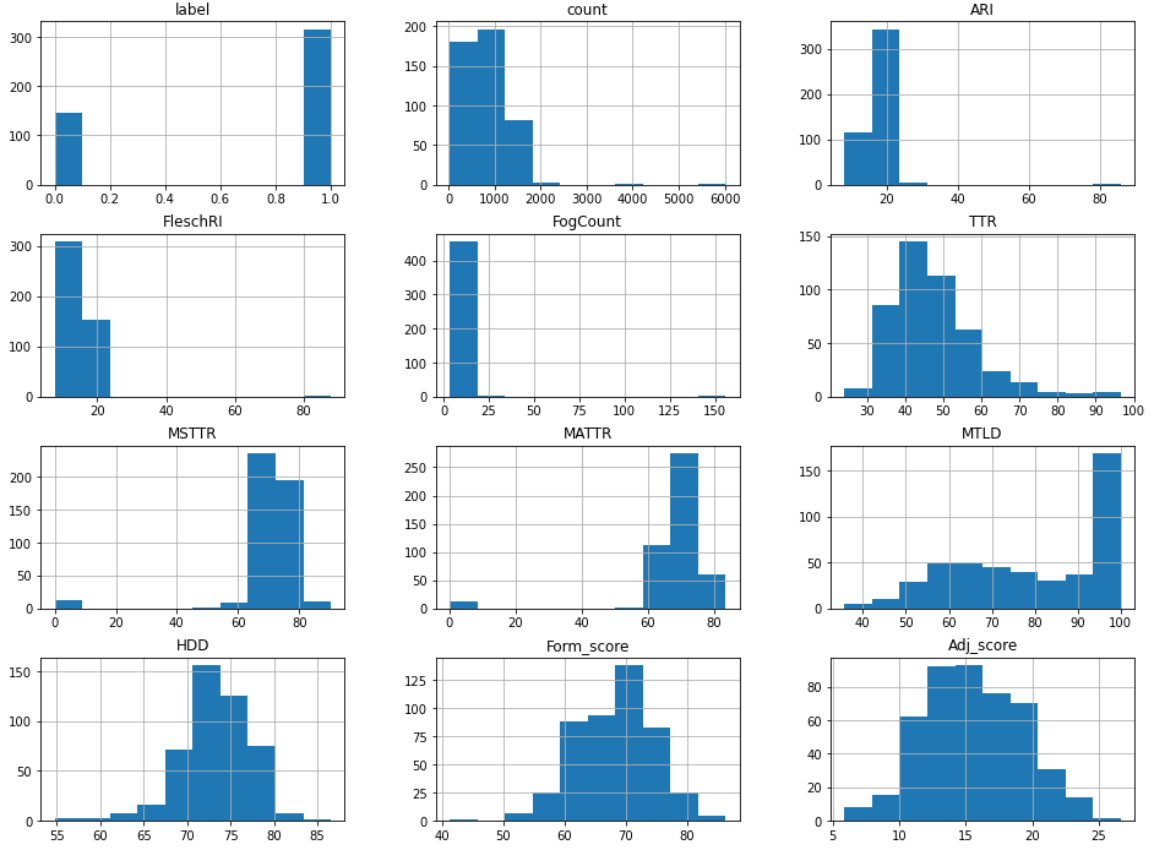


Figure 5.11: Histogram Metrics Model Stylometry

formation to train the classifier. As already discussed for the first model, it is important to check that the data is balanced before training the classifier. Machine learning classifiers strongly favour classes with a larger number of samples.

It is presented in the figure 5.11 the distribution of the features obtained for the radical magazines dataframe. By this way, it is known how the data is distributed, which type of article are more typical in the dataframe, and how many data of each category will be used to train the model.

As it can be seen in the figure 5.11, the resulting data ranges are vast, and in most cases, the data is not distributed. Therefore, normalization and other cleans are needed. Normalization is a technique, often applied as part of data preparation, for machine learning, which goal is to change the values of numeric columns to a common scale, without distorting differences in the ranges of values. This technique is usually required when features have different ranges, as it is this case [46]. On the other hand, the cleaning method has consisted on the removal of “dirty” values in some columns. As it can be seen in the ARI feature, there is one alone very big value in contrast with the others. Also, for the TTR and MSTTR, which present some rows with zero values. After normalizing and cleaning the dataframe,

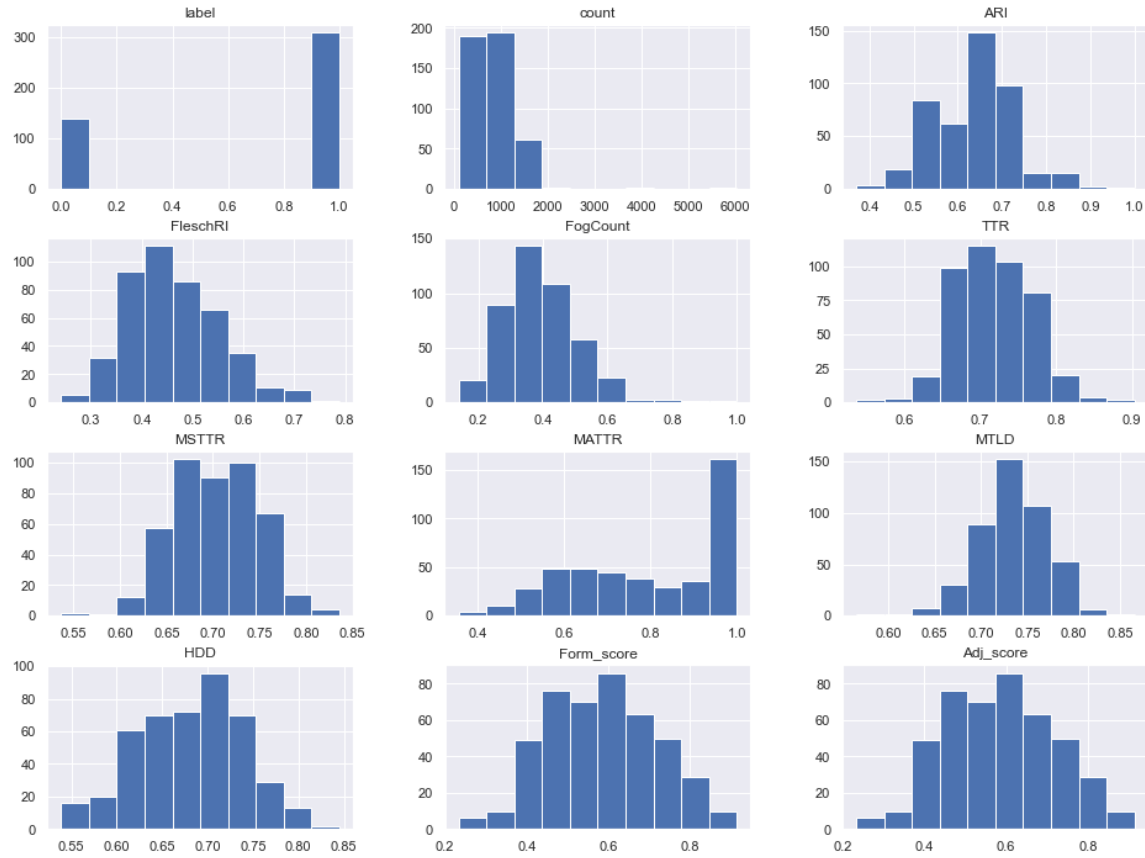


Figure 5.12: Histogram Metrics (Normalized) Model Stylometry

the figure 5.12 represents the new distribution.

Besides data distribution, it is also needed to represent the relation between the features. To do so, correlation is used. The figure 5.13 shows the correlation between the features and provides the information about which of them is most important. To understand this idea, it is needed to compare the correlation between the “propaganda label” and the other features. In this case, it can be seen that the most relevant feature for the decisions will be the MTLD metric.

As it can be seen, the MATTR and MSTTR features are the more relevant to classify a label between radical and non-radical. Selecting the MATTR, the figure 5.15 represents the differences in the distribution between non-radical and radical entries respectively.

### 5.3.4 Classification and Evaluation

In the same way than the model 1, the last step of the model 2 is to make it able to classify new articles with the higher accuracy as possible. This step consists in training a classification model and evaluate it. As it was explained in the first model, different

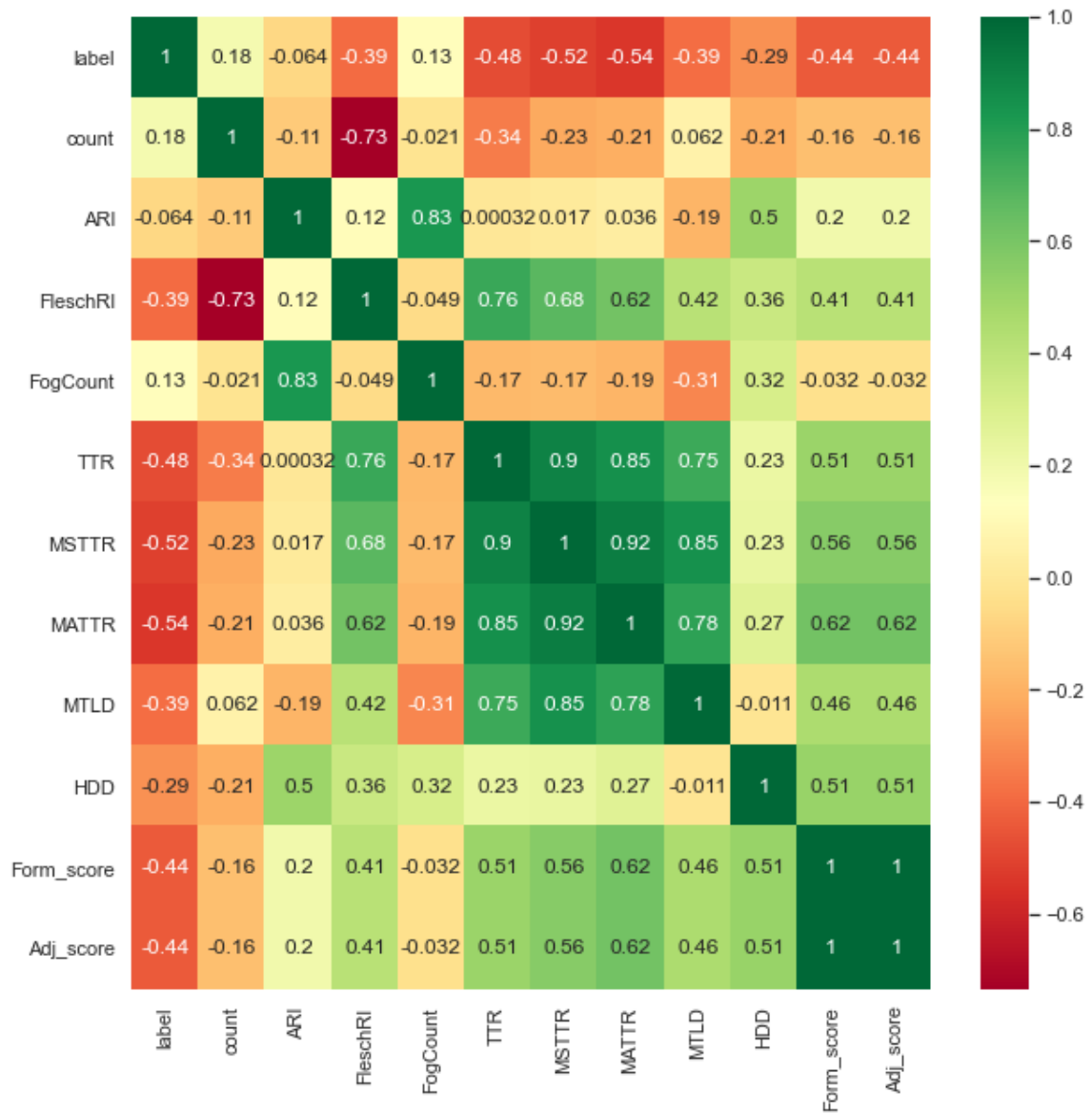


Figure 5.13: Metrics correlation Model Stylometry

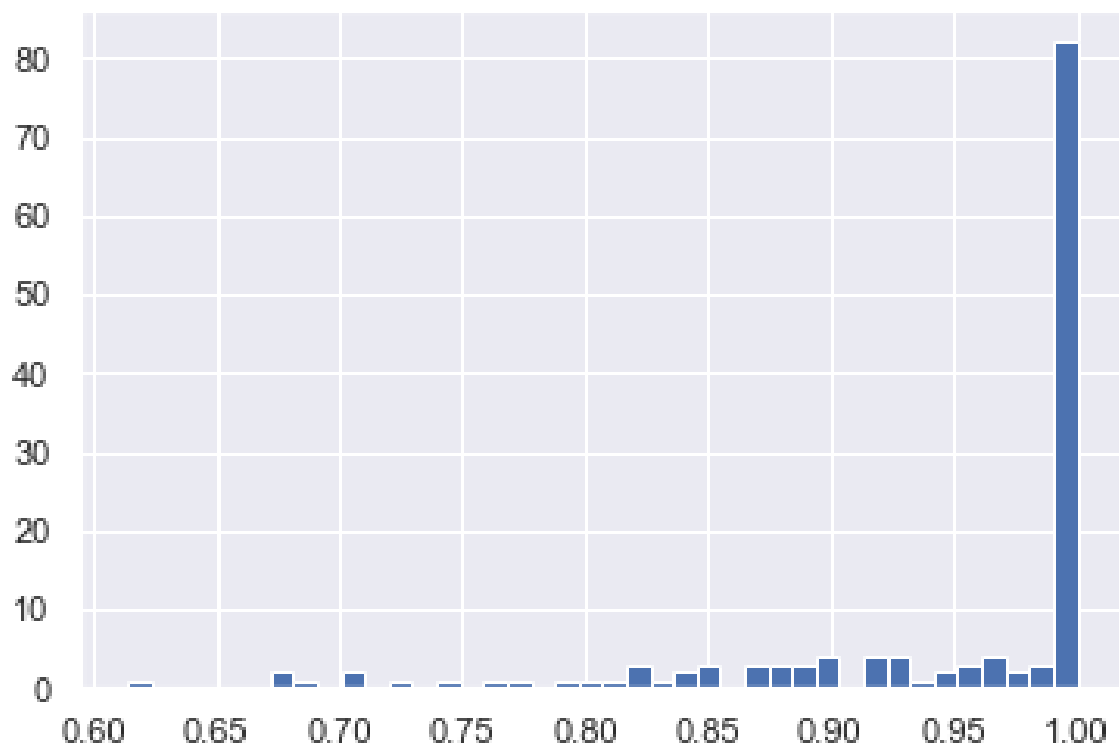


Figure 5.14: Histogram MATTR - Model Stylometry (label = 0)

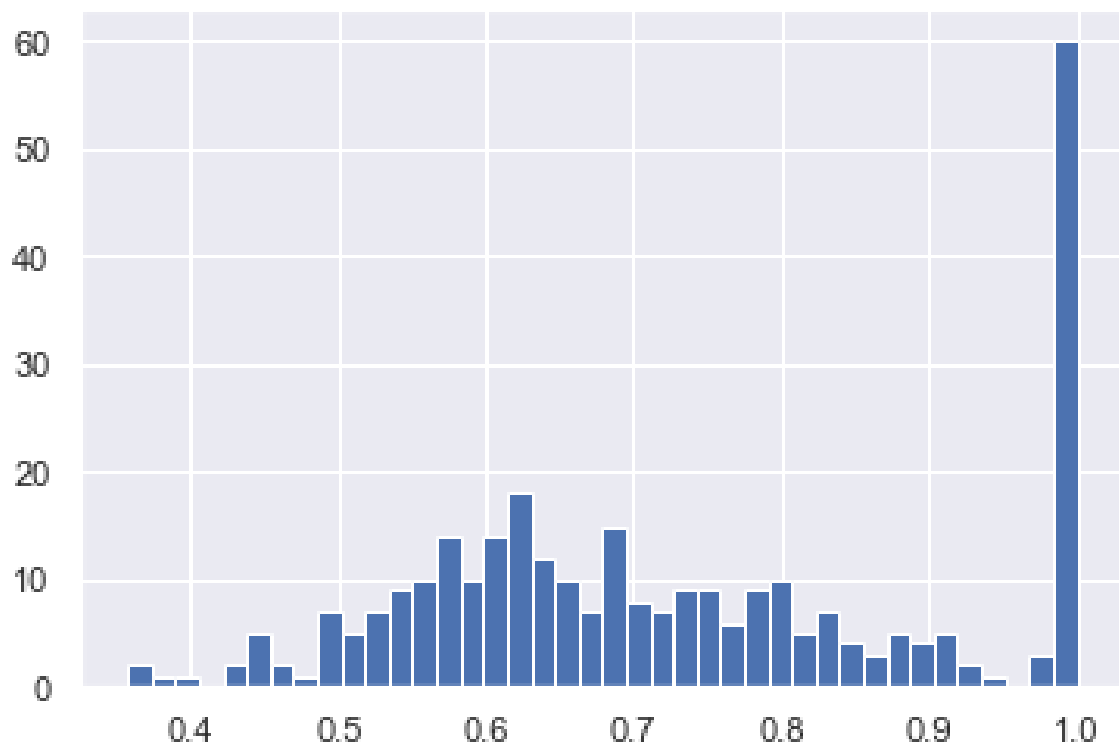


Figure 5.15: Histogram MATTR - Model Stylometry (label = 1)

	Dataframe used	Model used	Algorithm used	Balanced data	Balanced data method	Best Accuracy	Accuracy	Precision	Recall	f1_score
8	Proppy_dev	Sentiment Analysis	ExtraTreesClassifier	True	Under sampling	0.912	0.817204	0.855072	0.893939	0.874074
5	Proppy_dev	Sentiment Analysis	GradientBoostingClassifier	True	Under sampling	0.908	0.741935	0.818182	0.818182	0.818182
6	Proppy_dev	Sentiment Analysis	AdaBoostClassifier	True	Under sampling	0.898	0.741935	0.828125	0.803030	0.815385
1	Proppy_dev	Sentiment Analysis	RandomForestClassifier	True	Under sampling	0.896	0.827957	0.890625	0.863636	0.876923
2	Proppy_dev	Sentiment Analysis	DecisionTreeClassifier	True	Under sampling	0.868	0.752688	0.864407	0.772727	0.816000
0	Proppy_dev	Sentiment Analysis	LogisticRegression	True	Under sampling	0.810	0.752688	0.921569	0.712121	0.803419
7	Proppy_dev	Sentiment Analysis	KNeighborsClassifier	True	Under sampling	0.656	0.537634	0.744681	0.530303	0.619469
4	Proppy_dev	Sentiment Analysis	BernoulliNB	True	Under sampling	0.572	0.741935	0.818182	0.818182	0.818182
3	Proppy_dev	Sentiment Analysis	SVC	True	Under sampling	0.530	0.677419	0.860000	0.651515	0.741379

Figure 5.16: Best classifier for Stylometry Model

algorithms are going to be tested (the same as in the first model).

This model takes much more time to process the entries than the first one, as it needs to analyse many aspects of the text. Therefore, if the training propy dataframe is used, it will take near to two days to extract all the metrics for all the entries (35902 entries that match the requirement). As a solution, the radical dataframe has been used for this purpose. Extracting all the metrics takes 56 minutes. In fact, the graphs that were presented in the section 5.3.3, belong to these dataframes.

Once the features have been extracted, the model is trained. Then, the articles from the testing dataframes are passed through the model, and the output is compared to the real output of the dataframe.

As well as it was done in the first model, the best hyper-parameters need to be found for each algorithm. By using GridSearchCV, these parameters were found and the evaluation of the algorithm was done. The results obtained for each one is shown in the figure 5.16. The results are sorted by f1 score, and the best results are obtained for the “Extra Trees Classifier” algorithm. Specifically for this algorithm, the best hyper-parameters obtained were:

Best params: `{ 'clf__min_samples_split': 2, 'clf__random_state': 0, 'clf__max_depth': None, 'clf__n_estimators': 1000, 'clf__criterion': 'entropy' }`

Once it is known the algorithm and the balancing method that need to be used, it is the moment to implement the complete model, training it with the radical magazines dataframe, and testing it with the propy testing dataframe. In this case, in order to enable the possibility to test the model not only with the propy dataset but also with a set of data from the radical magazines dataset, it has been divided this last dataset into training and testing. The model was trained using an 80% of the radical magazines dataframe. By this way, a batch of data could be kept for also testing the model. Therefore, the results obtained are presented in the table 5.9. As it can be seen in the “radical magazines” columns, the values are the same. This is due to the model learns the same independently

Balancing method	Without balancing		Balancing with Random Over Sampler	
Tested with	Radical Magazines	Proppy test	Radical Magazines	Proppy test
Accuracy	0.827	0.535	0.827	0.539
Precision	0.867	0.519	0.867	0.52
Recall	0.894	0.96	0.894	0.962
F-score	0.88	0.674	0.88	0.676

Table 5.9: Model Stylometry results (trained with 80% of the data)

on whether the data is balanced or not. Furthermore, the results for the testing samples are pretty similar too.

Comparing the results without applying balancing and applying the Random Over Sampler, it can be seen that the results are pretty similar. The reason due this is happening is how the model is working. Applying the balancing method, as explained in the section 3.1.1.1, the new samples are randomly created taking into account the existing ones. Therefore, the articles style of each new sample will be the same of the already existing ones. However, it is useful to apply the balance in order to avoid the model just skip the features and predict the predominant class.

## 5.4 Evaluation

This section is presenting a general sum-up of the models, and the conclusions after calculating the results of both methods.

First of all, two natural language processing models have been developed. The first one is a similarity-based deep-learning model, and the second one is a stylometry based model. The main difference between them, is the features that are extracted to train the model. In the first case, the features are based on SIMON [85], and in the second case, the features are based on text style [21].

To obtain the results of table 5.10, the similarity-based DL model was trained with proppey dataset (total length of 35.902 samples), and the stylometry model was trained with radical magazines dataset (total length of 462 samples). Each of them has been tested with both, proppey and radical magazines datasets, to see how good the model is, predicting

data from a different dataset than the one used for training.

The first model is based on the Gradient Boost Classifier, and the best results were obtained by applying Under Sampling balancing method. On the other hand, the second model is based on the Extra Trees Classifier, and the best results were obtained applying Over Sampling balancing method. The differences of the classifiers used is just the kind of features extracted for each model. While the differences of the balancing method relies on the amount of data used for training.

At this point, it is reflected the importance of training the model with homogeneous data. The common point for both models is that, the results obtained, when predicting samples from the same dataset used for training, are much better than the ones obtained when predicting samples from the other dataset. The reason due to this is happening is just the way each dataset has been labelled.

On the one hand, the radical magazines dataset was labelled just looking the source the articles come from. In many cases, this is a very good approach, like, for example, when the authorship of a text want to be extracted. In many others, errors are more evident.

On the other hand, as it has been commented along the thesis, the propy datasets were labelled by using a technique known as distant supervision, which allows the generation of training data. This method generates a large amount of training data (most times a little bit noisy), with the particular exception that generating negative examples of a concept relation is more difficult than for a positive one.

Table 5.10 is presenting a comparison between the results obtained for each model. As it can be seen, the results obtained with the first model are better than the ones obtained with the second, in terms of accuracy and f-score, not only when testing with the same dataset, but also when testing with the different dataset. Therefore, the similarity-based model will be used in the pipeline for visualizing scraped data.

After the models are created, and once the best model is imported in the pipeline (section 4.1), the article scraping has been executed and presented in Kibana (section 4.5. Figure 5.17 represents some of the most common words by source (where the article has been extracted from) for those articles that has been labeled as “radical”.

Other case of use could be, to inspect which web is publishing more articles tagged as radical. For this case, figure 5.18 represents the top five web news that have published more articles labeled as radical in the last month (from May the 3<sup>rd</sup> to June the 3<sup>rd</sup>). In the figure, the value “others” is predominating due to there are a total of 248 sources, and the ones not appearing are grouped. Anyway, having almost 30 articles tagged as radical, is a high proportion for just one month.

	Similarity-based DL model		Stylometry model	
Tested with	Proppy test	Radical Magazines	Proppy test	Radical Magazines
Accuracy	0.847	0.724	0.539	0.827
Precision	0.978	0.845	0.52	0.867
Recall	0.847	0.725	0.962	0.894
F-score	0.907	0.780	0.676	0.88

Table 5.10: Comparison of models' results

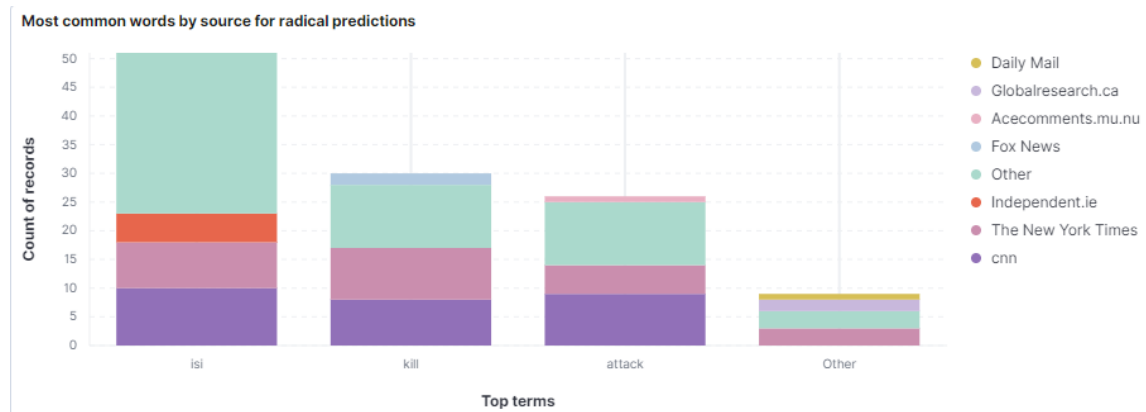


Figure 5.17: Common words by source (radical articles)

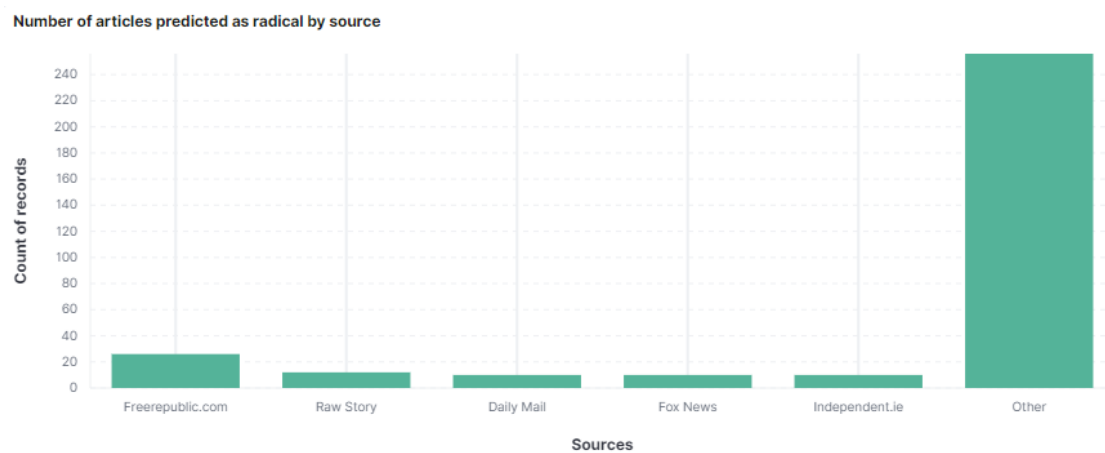


Figure 5.18: Top 5 sources with more radical articles

## Conclusions

---

*In this chapter the conclusions extracted from the project are presented. In section 6.1, the main conclusions extracted from the thesis development are commented. Then, the section 6.2 presents the achieved goals are mentioned, taking into account the objectives established at the beginning of the thesis.*

### 6.1 Conclusions

To conclude, this section summarizes the main concepts that have been explained in the previous sections of the thesis.

The great evolution that technology has undergone over the last few decades has led to social networks and digital media becoming the main means of communication worldwide. The fast growth suffered has also led to loss the control not only on the type of content is published but also in the great vast of people publishing in different websites. On the one hand, this can favour the knowledge sharing and the ideas spread, which are two things pretty good for society development. On the other hand, there are also many people who take advantages of the situation and misuse the media with the objective of promoting hate or extremist ideology radicalization.

There are so many different ways to fight against the social media misuse. Among these options, it is highlighted the development of Natural Language Processing and Machine Learning techniques. Using these techniques, it is possible to predict with a certain prob-

ability, whether an article, text, or new published on the internet, is considered radical propaganda or not. Furthermore, the development of this techniques also allows to automatize the surveillance processes of detecting fraudulent publications. Therefore, one of the main goals of the thesis has been the development of a model able to predict if a text extracted from public newspapers is considered radical propaganda or not.

To enable the possibility to develop a model, it is firstly necessary to correctly define the problem. There are several variables that need to be considered. Among these variables, the language of the texts and the subject about the texts should be, are two of the main ones. For the thesis, the matter has been the dissemination of radical Islamic propaganda on texts in English. Besides, there are a lot of possibilities of creating models. Therefore, it has been necessary to carry out a large study on different algorithms, balancing methods and pre-processing techniques, to determine which were the best conditions for each model.

Once the problem definition is clarified, a solution has to be proposed. Following the advantages already commented, it has been developed not one but two Natural Language Processing models to predict whether a text is considered Islamic radical propaganda or not. On the one hand, the first model developed is based on Similarity, which is basically based on feature extraction to determine whether a text expresses a radical or non-radical idea. On the other hand, the second model is based on Stylometry, which is basically the application of the study of linguistic style to a text to extract linguistic features. Each of the models has had independent developments and results. The table 6.1 represents the results obtained for each model. As it can be seen, the results obtained for the first model are better than the results obtained for the second model. It is also highlighted the difference when predicting texts from the same dataset than from the different one. This difference comes from the way each dataset has been labelled. As commented along the memory, the Propopy datasets were labelled by using a technique known as distant supervision. On the other hand, the Radical Magazines dataframe was labelled manually, depending on the source the article was extracted from, considering radical all articles extracted from Rumiya and Dabiq, and non-radical all the articles extracted from The New York Times and CNN. Therefore, it is actually more accurate the model trained with the Propopy dataset.

However, a batch of results without a correct visualization is useless. Therefore, a dashboard has been also implemented to represent graphically the results obtained. To develop the dashboard, many technologies like Docker and Luigi has been used. In this sense, a pipeline of tasks managed by Luigi has been developed to automatically deploy containers using Docker and Docker-compose, where other technologies run. The pipeline has three main steps, for which a specific container is used in each case. Data extraction is based on the GSI Crawler, which extracts articles from The New York Time, the CNN, and many other public newspapers of Google News. Data Processing is based on the models

	Similarity-based DL model		Stylometry model	
Tested with	Proppy test	Radical Magazines	Proppy test	Radical Magazines
Accuracy	0.847	0.724	0.539	0.827
Precision	0.978	0.845	0.52	0.867
Recall	0.847	0.725	0.962	0.894
F-score	0.907	0.780	0.676	0.88

Table 6.1: Conclusion - comparison of models' results

created. A model can be imported in the data processing task in order to extract the features of a text and therefore, predict if the text is radical or not. Finally, data storage is based on Elasticsearch databases. Using index to store each extraction is possible to later represent the data with Kibana, an intuitive dashboard to create representative figures of the stored data.

## 6.2 Achieved Goals

Starting from the project goals defined in the section 1.2, the following are the achieved objectives.

- G1 – Initial Goal: Deepen in the use of NLP/ML techniques
  - Fulfilled: **YES**
  - Comments: Despite having a basic knowledge at the starting point of the thesis, the development of not one but two different models based on different techniques has helped a lot in acquiring this further knowledge about NLP and ML techniques.
- G2 – Initial goal: Develop at least one NLP model, which takes as input English texts and predict samples with, at least, an 85% of accuracy from texts coming from the same source, and a 70% of accuracy from different sources.
  - Fulfilled: **YES**
  - Comments: Not one but two different models have been developed. The first one based on Similarity and the second one based on Stylometry. Both takes as

	F-score goal	F-score with Similarity	F-score with Stylometry
Same dataset	85%	90.7%	88%
Different dataset	70%	78%	67.6%

Table 6.2: Summarized results of the models

input English texts, both fulfill the f-score goal set for texts of the same dataset, and only the first one fulfilled the f-score goals for text comings from different sources. The results of each model is compared with the goal in the table 6.2.

- G3 – Initial goal: Include customized models in the GSI pipeline, and introduce a new source of scraping.
- Fulfilled: **YES**
- Comments: The model that has been introduced in the GSI pipeline is the Similarity-based one, which is the one for which the results were better. To introduce it, it has been necessary first to develop, train and export the model, to export the features extractor, and to import both in the pipeline. Later, each result that is scraped, is analysed and labelled automatically by the model. In addition, a new scraping source has been configured, which is Google News.
- G4 – Initial goal: promote the awareness of people about radicalization.
- Fulfilled: **Yes**
- Comments: The development and the publication of the thesis may aware people who read it about the importance to continue developing natural language processing tools, not only to countermeasure Islamic radicalization, but also to make better the machine human-writing understanding. This goal cannot be considered as fulfilled after the publication of the thesis. At the moment this section is published, the goal is automatically fulfilled.

## Impact of this project

---

This section contains the analysis of the impact of the realization of this project from a social, economic, environmental and ethical point of view.

### **A.1 Social impact**

As commented along the thesis, social media and social information has grown a lot during the last years, and it is expected to continue growing for the next ones. Social networks are the main source of information nowadays. Many times, due to lack of digital education, to the easiness to hide your identity, and the people reachable from the internet, this media is misused or used with harmful purposes.

On the one hand, the development of a modern web tool to filter messages, articles, news, or whatever is going around the net by text, could be useful to avoid this kind of abuses. The possibility to interact with a pipeline to apply different Natural Language Processing models, allows tackling problems from different situations.

On the other hand, the chosen theme could point to a very particular group of people, on which many times is included all the people who belong to the Islamic religion. This fact may harm some of these people who supports Islamism, but does not support Islamic state or radical groups.

Despite of what many people could think, the development of new tools, frameworks, technologies, or whatever improves the security and well-being of the society, needs to be

considered as an advance, not as a weapon that hurts people. In fact, this project just analysed published texts without making any bad to society.

### A.2 Economic impact

The development of this project has not involved any costs, as it has been used the current infrastructure managed by the intelligent system group (GSI) of the university. If the solution wants to be distributed to non-public organizations, or to public organizations outside university environment, it would include some costs related to the server where is running the enabling technologies (ElasticSearch, Kibana, models and crawlers).

Taking into account the advantages this solution provides, such as the efficiency improvement, the reduction of time waste and effort, the comfort and ease of use, and assuming that this tool is automatic (no people is needed), it can be concluded that the overall costs could be reduced.

### A.3 Environmental impact

The main element that needs to be taken into account in this section, is the energy consumed by the equipment that keeps the project working. This equipment includes the computers, the servers and any other device that could be used. The energy consumed during a whole year could be reduced by hiring infrastructure providers, or by feeding the system with renewable energies. Besides, it can be considered the use of green computing models [67], which are expected to enhance the environmental impact of the project.

### A.4 Ethical impact

The first ethical implication of the project is related to the Islamic religion, which could be considered attacked as being analysed many texts related to it. However, far from attacking the religion, this thesis wants to provide an example about what can be achieved by using natural language processing models in radicalism detection.

The second ethical problem could be the exposition of articles authors, which could be also considered a privacy issue. Actually, the articles are published on online magazines or newspapers, so everyone could read them. What could lead to the ethical problem is the fact to categorize a text as radical or not by using a software tool. If many texts that belong to the same author are considered radical, probably the author will be marked as ISIS supporter. Due to this issue, the authors of the texts are discarded in the preprocessing

step, so the final dashboard is not representing them. By this way, any privacy issue will be avoided.

If the sources want to be extended with social networks like Twitter or Facebook, there need to be also considered the policies from the companies these brands belong to.



## Economic budget

---

In this appendix it is shown the cost analysis involved in this project, including effort, materials and economic aspects.

### **B.1 Project structure**

The project is mainly divided in three parts. The first one is the developing of the models, the second one is the development of the dashboard, and the third one is the report writing. Each of these tasks can be also divided into subtasks, which are presented in the table B.1, including also an estimation of the hours that each one has taken. Therefore, it can be assured that the total duration of this project has been **143,5** days, which, estimating that each day has a mean of 5 working hours, it is traduced as **717,5** hours. Taking into account that 1 ECTS is the same to 25h of work, and that the thesis has 30 ECTS, the estimated needed time to develop the thesis is 750 hours, which is more or least the time dedicated.

Main task	Subtasks	Time (days)
Models' development		50
	Similarity-based DL Model	30
	Model Stylometry	20
Dashboard		34
	Learning	8
	Development	22
	Testing	4
Report writing		54,5
	Chapter 1 - Introduction	1
	Chapter 2 - State of Art	10
	Chapter 3 - Enabling technologies	18
	Chapter 4 - Architecture	8
	Chapter 5 - Classification models	15
	Chapter 6 - Conclusion	0,5
	Appendixes	2
Back office work		7
	Documentation (already included)	0
	Meetings	3
	Corrections	2
<b>Total days</b>		<b>143,5</b>

Table B.1: Project structure

## B.2 Physical resources

The project has been developed in a personal laptop. In addition, it has been used the infrastructure provided by the Intelligent System Group (GSI). The requirements of the project are not too restrictive. However, the more powerful equipment, the faster the tests go, and more test could be done then. In this case, the personal laptop has the following characteristics:

- CPU: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
- RAM: 16GB
- Disk: 326GB (from 916GB the laptop originally had)

Actually, the laptop is very good, but it is also a little bit old, so some times it has given some problems during the development of the thesis. In addition to these resources, it has been used the cluster of the GSI group, which, in accordance to the specifications, it consists in a master node and two workers, which are configured as follows:

- DELL PowerEdge R320
- Intel® Xeon® E5-2430 v2 (2,50GHz, 6 cores with hyperthreading, 15M cache, QPI 7,2GT/s, Turbo) 80W
- 4x32GB RDIMM, 1333 MHz
- 3x3TB, SATA, 7,2k rpm, 3,5" hot-swappable. RAID+LVM. About 3 TB are currently shared by all nodes using glusterfs.

There is no need of any other software, neither of any other license, so, according to the specifications aforementioned, the estimated prices for both equipment are 1.000€ for the computer, and 2.600€ for the cluster. Obviously, the cluster has not been bought for the project, but in order to simply calculations, it will be considered as a total cost. Therefore, the **total physical resources cost is 3.600€**.

## B.3 Human resources

For the development of this project, it can be considered the work of one person during all the project, and the consult to an expert in specific points of the development. Taking into account that the person working in the project is an engineering graduate, the estimated salary is 20.600€/year [47], which traduced in hours (1.826 annual working hours)

Annual salary		20.600€
IRPF	12%	2.472€
Social security contributions	6,4%	1.308,1€
Annual salary		20.600€
Annual working hours		1.826
Salary/hour		9,21€
Worked hours		717,5
Cost		6.609,13€
Expert cost/h		50€
Expert hours		10
Cost		500€
Total cost		7.109,13€

Table B.2: Human resources costs

is 11,28€/hour. However, if we discount taxes and fees, the net salary is 16.819,9€, which is traduced in 9.21€/hour. Besides, the expert estimated cost is 50€/hour. As the project has lasted 717.5 hours, and 10 of those hours have been with an expert, it is considered that the **total cost of the development is 7.109,13€**.

## B.4 Conclusion

Grouping all the results presented in the sections above, the total cost of the project is **10.709,13€**. Taking into account that the project has had a total duration of around 7 months, this means that the monthly cost has been of 1.492,56 €. The development of this project in any other private business may involve other costs related to licenses that in this

Asset	Price
Personal laptop	1.000€
Cluster	2.600€
Human resources	7.109,13€
<b>Total costs</b>	<b>10.709,13€</b>

Table B.3: Total costs of the project

project has not been considered, as they are free-use for university. Therefore, in conclusion, I believe that it is good that the university carries out this type of research project, as it promotes the development of technology, as in this case of NLP, while the university gains even more prominence in the technology sector.



# Bibliography

---

- [1] The Luigi Authors Revision 54a34736. Luigi - getting started. <https://luigi.readthedocs.io/en/stable/>, 2020.
- [2] S. Ahmad, M.Z. Asghar, and F.M. Alotaibi. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. <https://doi.org/10.1186/s13673-019-0185-6>, 2019.
- [3] Ahmed Al-Rawi. Video games, terrorism, and isis’s jihad 3.0. *Terrorism and Political Violence*, 30, 08 2016.
- [4] Mohamed Alloghani, Dhiya Al-Jumeily Obe, Jamila Mustafina, Abir Hussain, and Ahmed Aljaaf. A systematic review on supervised and unsupervised machine learning algorithms for data science, 01 2020.
- [5] Mohamed Alloghani, Dhiya Al-Jumeily Obe, Jamila Mustafina, Abir Hussain, and Ahmed Aljaaf. A systematic review on supervised and unsupervised machine learning algorithms for data science, 01 2020.
- [6] Altexsoft. Data science vs machine learning vs ai vs deep learning vs data mining: Know the differences. <https://www.altexsoft.com/blog/data-science-artificial-intelligence-machine-learning-deep-learning-data-mining> January 2021. Published on Altexsoft.
- [7] Altexsoft. Unsupervised learning: Algorithms and examples. <https://www.altexsoft.com/blog/unsupervised-machine-learning/>, April 2021.
- [8] Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246, 2017.
- [9] Michael Ashcroft, Ali Fisher, Lisa Kaati, Enghin Omer, and Nico Prucha. Detecting jihadist messages on twitter. In *2015 European Intelligence and Security Informatics Conference*, pages 161–164, 2015.
- [10] Luis Bahillo. Historia de internet: cómo nació y cuál fue su evolución. <https://marketing4ecommerce.net/historia-de-internet/>, May 2021. Published in Marketing4ecommerce.
- [11] Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. Propy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864, 2019.

- [12] Cristina Bayas. As journalism evolves, it seems harder to silence dissent. <https://mcristinabayas.wordpress.com/tag/evolution-of-media/>, March 2016.
- [13] Hoss Belyadi and Alireza Haghighat. Chapter 5 - supervised learning. In Hoss Belyadi and Alireza Haghighat, editors, *Machine Learning Guide for Oil and Gas Using Python*, pages 169–295. Gulf Professional Publishing, 2021.
- [14] Doosje Bertjan, M Moghaddam Fathali, W Kruglanski Arie, de Wolf Arjan, Mann Liesbeth, and R Feddes Allard. Terrorism, radicalization and de-radicalization. *Current Opinion in Psychology*, 11:79–84, 2016. Intergroup relations.
- [15] Umesh .A Bhat. Log analytics with deep learning and machine learning. <https://hackernoon.com/log-analytics-with-deep-learning-and-machine-learning-20a1891ff70e>, May 2017.
- [16] Jason Browlee. A gentle introduction to threshold-moving for imbalanced classification. [https://imbalanced-learn.org/stable/references/generated/imblearn.under\\_sampling.NearMiss.html](https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.NearMiss.html), February 2020.
- [17] Elasticsearch B.V. Welcome to elastic docs. <https://www.elastic.co/guide/index.html>, 2022.
- [18] Mashrur Chowdhury, Amy Apon, and Kakan Dey. Machine learning in transportation data analytics. <https://www.sciencedirect.com/topics/psychology/machine-learning>, 2017. Published on Elsevier.
- [19] Mattia Cinelli. A tutorial on luigi, the spotify’s pipeline. <https://towardsdatascience.com/a-tutorial-on-luigi-spotifys-pipeline-5c694fb4113e>, July 2020.
- [20] Michael A. Covington and Joe D. McFall. Cutting the gordian knot: The moving-average type-token ratio (mattr). *Journal of Quantitative Linguistics*, 17(2):94–100, 2010.
- [21] Álvaro de Pablo Marsal, Óscar Araque Iborra, and Carlos Ángel Iglesias Fernández. Radical text detection based on stylometry. In *Proceedings of 6th International Conference on Information Systems Security and Privacy*, pages 1–8, Febrero 2020.
- [22] Brian Dean. How many people use twitter in 2022? [new twitter stats]. <https://backlinko.com/twitter-users#twitter-statistics>, January 2022. Published in Backlinko.
- [23] Angelica Lo Duca. How to balance a dataset in python. <https://towardsdatascience.com/how-to-balance-a-dataset-in-python-36dff9d12704>, March 2021. Published in Towards Data Science.
- [24] IBM Cloud Education. Strong ai. <https://www.ibm.com/cloud/learn/strong-ai>, August 2020. Published on IBM Cloud Learn Hub.
- [25] Daniel Eriksson. Using the f-measure to test formality in sports reporting. <http://www.diva-portal.org/smash/get/diva2:1223014/FULLTEXT01.pdf>, 2017.

- 
- [26] Alex Chengyu Fang and Jing Cao. Adjective density as a text formality characteristic for automatic text classification: A study based on the british national corpus. <https://aclanthology.org/Y09-1015.pdf>, 2009.
- [27] Tobias Feakin and Benedict Wilkinson. The future of jihad: What next for isil and al-qaeda? Technical report, Australian Strategic Policy Institute, 2015.
- [28] Yúbal Fernández. 47 páginas .onion para visitar el lado amable de la deep web. <https://www.genbeta.com/web-20/47-paginas-onion-para-visitar-el-lado-amable-de-la-deep-web>, January 2021. Published on Xataka.
- [29] Yúbal Fernández. Deep web, dark web y darknet: éstas son las diferencias. <https://www.xataka.com/servicios/deep-web-dark-web-darknet-diferencias>, April 2021. Published on Xataka.
- [30] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 11 1997.
- [31] Vijay Gadepally, Justin Goodwin, Jeremy Kepner, Albert Reuther, Hayley Reynolds, Siddharth Samsi, Jonathan Su, and David Martinez. Ai enabling technologies. [https://vijayg.mit.edu/sites/default/files/images/EnablingTechnologies\\_042319.pdf](https://vijayg.mit.edu/sites/default/files/images/EnablingTechnologies_042319.pdf), April 2019. Lincoln Laboratory - MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT).
- [32] Isaac Galatzer-Levy, Kelly Ruggles, and Zhe Chen. Data science in the research domain criteria era: Relevance of machine learning to the study of stress pathology, recovery, and resilience. *Chronic Stress*, 2:247054701774755, 01 2018.
- [33] Gartner. Gartner predicts the future of ai technologies. <https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-ai-technologies>, November 2019. Published on Gartner.
- [34] Gartner. Gartner forecasts global spending on wearable devices to total 81.5 billion dollars in 2021. <https://www.gartner.com/en/newsroom/press-releases/2021-01-11-gartner-forecasts-global-spending-on-wearable-devices-to-total-81-5-billion-in-2021>, March 2022.
- [35] Fawaz A. Gerges. The world according to isis. <https://www.foreignpolicyjournal.com/2016/03/18/the-world-according-to-isis/>, March 2016. Published on The Foreign Policy Journal.
- [36] JOHN GRAMLICH. 10 facts about americans and facebook. <https://www.pewresearch.org/fact-tank/2021/06/01/facts-about-americans-and-facebook/>, June 2021. Published in Pew Research Center.
- [37] Rohit Gupta. A machine learning investment thesis. <https://medium.com/rohits-perspectives/a-machine-learning-investment-thesis-4e6bdf66d07d>, September 2018.

## BIBLIOGRAPHY

---

- [38] Stevan Harnard. The turing test is not a trick: Turing indistinguishability is a scientific criterion, 1992.
- [39] Francis Heylighen, Jean Marc Dewaele, and Léo Apostel. Formality of language: definition, measurement and behavioral determinants. In *Formality of Language: definition, measurement and behavioral determinants*, 1999.
- [40] ADAM HUGHES and STEFAN WOJCIK. 10 facts about americans and twitter. <https://www.pewresearch.org/fact-tank/2019/08/02/10-facts-about-americans-and-twitter/>, August 2019. Published in Pew Research Center.
- [41] Arman Hussain. K-nearest neighbors (knn) and its applications. [https://medium.com/@arman\\_hussain786/k-nearest-neighbors-knn-and-its-applications-7891a4a916c6](https://medium.com/@arman_hussain786/k-nearest-neighbors-knn-and-its-applications-7891a4a916c6), July 2020.
- [42] imbalanced-learn developers. Imbalanced-learn documentation. [https://imbalanced-learn.org/stable/user\\_guide.html#user-guide](https://imbalanced-learn.org/stable/user_guide.html#user-guide), 2014-2022.
- [43] SAS Institute Inc. Big data: What is and why it matters. [https://www.sas.com/en\\_us/insights/big-data/what-is-big-data.html#history](https://www.sas.com/en_us/insights/big-data/what-is-big-data.html#history), 2022. Published in SAS.
- [44] Immune Technology Institute. De la prueba de turing al móvil: así se ha ido metiendo la ia en nuestras vidas durante más de medio siglo. <https://www.xataka.com/n/prueba-turing-al-movil-asi-se-ha-ido-metiendo-ia-nuestras-vidas-durante-medio-siglo> July 2020. Published on Xataka.
- [45] Mansoor Iqbal. Tiktok revenue and usage statistics (2022). <https://www.businessofapps.com/data/tik-tok-statistics/>, February 2022. Published in Business of Apps.
- [46] Urvashi Jaitley. Why data normalization is necessary for machine learning models. <https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029>, October 2018.
- [47] Jobted. Sueldo del ingeniero de telecomunicaciones en españa. <https://www.jobted.es/salario/ingeniero-telecomunicaciones>, 2021. Published in Jobted.
- [48] Tomcy John and Pankaj Misra. Data lake for enterprises. <https://www.oreilly.com/library/view/data-lake-for/9781787281349/>, May 2017. Published in O'reilly.
- [49] Miriam Johnson. Timeline of social media, 2021. <https://www.booksaresocial.com/timeline-of-social-media-2021/>, June 2021.
- [50] Elizabeth Susan Joseph. Random state meaning in balancing methods. <https://stackoverflow.com/questions/28064634/random-state-pseudo-random-number-in-scikit-learn>, January 2015.

- [51] Armaan Kaur, Jaspal Kaur Saini, and Divya Bansal. Detecting radical text over online media using deep learning. *arXiv preprint arXiv:1907.12368*, 2019.
- [52] Ajitesh Kumar. Accuracy, precision, recall & f1-score – python examples. [https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/#What\\_is\\_Accuracy\\_Score](https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/#What_is_Accuracy_Score), April 2022.
- [53] Nitish Kumar. Introduction to support vector machines (svms). <https://www.marktechpost.com/2021/03/25/introduction-to-support-vector-machines-svms/>, March 2021.
- [54] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Library to balance a dataset in python - nearmiss. [https://imbalanced-learn.org/stable/references/generated/imblearn.under\\_sampling.NearMiss.html](https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.NearMiss.html), August 2014.
- [55] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Library to balance a dataset in python - random over sampling. [https://imbalanced-learn.org/stable/references/generated/imblearn.over\\_sampling.RandomOverSampler.html](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html), August 2014.
- [56] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Library to balance a dataset in python - random under sampler. [https://imbalanced-learn.org/stable/references/generated/imblearn.under\\_sampling.RandomUnderSampler.html](https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html), August 2014.
- [57] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Library to balance a dataset in python - smote. [https://imbalanced-learn.org/stable/references/generated/imblearn.over\\_sampling.SMOTE.html](https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html), August 2014.
- [58] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.
- [59] Nelson Leonard. The deep web, the dark web, and simple things. <https://medium.com/@smartrac/the-deep-web-the-dark-web-and-simple-things-2e601ec980ac>, August 2017.
- [60] Qiong Liu and Ying Wu. Supervised learning, 01 2012.
- [61] Philip Mccarthy and Scott Jarvis. Mtd, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42:381–92, 05 2010.
- [62] Tom M. Michel. Machine learning. <https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf>, March 1997. Published on McGraw-Hill Science/Engineering/Math.
- [63] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*, pages 300–314, 2012.

- [64] Kasey Panetta. Cios can separate ai hype from reality by considering these areas of risk and opportunity. <https://www.gartner.com/smarterwithgartner/the-cios-guide-to-artificial-intelligence>, February 2019. Published on Gartner.
- [65] European Parliament and the Council of the European Union. Gdpr - personal data. <https://gdpr-info.eu/issues/personal-data/>, April 2016.
- [66] European Parliament and the Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>, April 2016.
- [67] Prof. Dr. Girish Kumar Patnaik, Miss. Bhagyashree, P. Lokhande, and Mr. Akshay G. Mahajan. Green computing metrics, methods and models. <https://www.ijert.org/research/green-computing-metrics-methods-and-models-IJERTV3IS031900.pdf>, March 2014.
- [68] Elisa Pont. Sunismo y chiismo, las dos ramas del islam. <https://www.lavanguardia.com/vida/junior-report/20190528/462436747454/sunitas-chiitas-islam.html>, November 2019. Published on La Vanguardia.
- [69] RAE. Definition of ideology by rae. <https://dle.rae.es/ideolog%C3%ADa>, 2021. Published on RAE.
- [70] RAE. tecnología. <https://dle.rae.es/tecnología>, March 2022.
- [71] Sebastian Raschka. Naive bayes and text classification i - introduction and theory, 2014.
- [72] scikit-learn developers. DictVectorizer. [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.DictVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html), 2007 - 2022.
- [73] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424, 1980.
- [74] Scott Shane and Ben Hubbard. Isis displaying a deft command of varied media. <https://www.nytimes.com/2014/08/31/world/middleeast/isis-displaying-a-deft-command-of-varied-media.html>, August 2014. Published on the Ney York Times.
- [75] Terence Shin. A mathematical explanation of naive bayes in 5 minutes. <https://towardsdatascience.com/a-mathematical-explanation-of-naive-bayes-in-5-minutes-44adebcdb5f8>, June 2020.

- 
- [76] Dima Shulga. 5 reasons why you should use cross-validation in your data science projects. <https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163> September 2018. Published in Towards Data Science.
  - [77] Simplilearn. The complete guide to machine learning steps. <https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps>, November 2021.
  - [78] Katy Stalcup. Aws vs azure vs google cloud market share 2021: What the latest data shows. <https://www.parkmycloud.com/blog/aws-vs-azure-vs-google-cloud-market-share/>, March 2022. Published in ParkMyCloud.
  - [79] Statista. Statista. <https://es.statista.com/grafico/17734/cantidad-real-y-prevista-de-datos-generados-en-todo-el-mundo/>, 2022.
  - [80] NLTK Team. Nltk documentation. <https://www.nltk.org/>, 2022.
  - [81] Pandas Development Team. Pandas documentation. <https://pandas.pydata.org/docs/>, April 2022.
  - [82] Himanshu Tripathi. What is balanced and imbalanced dataset? <https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5>, September 2019.
  - [83] Maryville university. The evolution of social media: How did it begin, and where could it go next? <https://online.maryville.edu/blog/evolution-social-media/>, 2022. Published in Maryville university blog.
  - [84] Unknown. La formación y desarrollo de las redes sociales – social media timeline 2021. <https://mdigital.com.vn/su-hinh-thanh-va-phet-trien-cua-mang-xa-hoi-social-media-timeline-2021/>, November 2021.
  - [85] Intelligent System Group UPM. Gsitr documentation. <https://gsi.upm.es/software/projects/gsitk/>, 2018.
  - [86] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. [https://radimrehurek.com/lrec2010\\_final.pdf](https://radimrehurek.com/lrec2010_final.pdf).
  - [87] Felipe Veloso. Clustering – k-nearest neighbors. <https://www.feedingthemachine.ai/clustering-k-nearest-neighbors/>, June 2019.
  - [88] Matteo Vergani and Ana-Maria Bliuc. The language of new terrorism: Differences in psychological dimensions of communication in dabiq and inspire. *Journal of Language and Social Psychology*, 37:0261927X1775101, 01 2018.
  - [89] Wikipedia. Automated readability index. [https://en.wikipedia.org/wiki/Automated\\_readability\\_index](https://en.wikipedia.org/wiki/Automated_readability_index), September 2021.

## BIBLIOGRAPHY

---

- [90] Wikipedia. Youden's j statistic. [https://en.wikipedia.org/wiki/Youden%27s\\_J\\_statistic](https://en.wikipedia.org/wiki/Youden%27s_J_statistic), October 2021.
- [91] Wikipedia. Big data. [https://en.wikipedia.org/wiki/Big\\_data](https://en.wikipedia.org/wiki/Big_data), February 2022. Published in Wikipedia.
- [92] Wikipedia. Dabiq (magazine). [https://en.wikipedia.org/wiki/Dabiq\\_\(magazine\)](https://en.wikipedia.org/wiki/Dabiq_(magazine)), April 2022.
- [93] Wikipedia. Gunning fog index. [https://en.wikipedia.org/wiki/Gunning\\_fog\\_index](https://en.wikipedia.org/wiki/Gunning_fog_index), January 2022.
- [94] Wikipedia. Islamism. [https://en.wikipedia.org/wiki/Islamism#Moderate\\_Islamism](https://en.wikipedia.org/wiki/Islamism#Moderate_Islamism), March 2022. Published on Wikipedia.
- [95] Wikipedia. List of terrorist incidents linked to the islamic state. [https://en.wikipedia.org/wiki/List\\_of\\_terrorist\\_incidents\\_linked\\_to\\_the\\_Islamic\\_State](https://en.wikipedia.org/wiki/List_of_terrorist_incidents_linked_to_the_Islamic_State), February 2022.
- [96] Wikipedia. Logistic regression - applications. [https://en.wikipedia.org/wiki/Logistic\\_regression#Applications](https://en.wikipedia.org/wiki/Logistic_regression#Applications), April 2022.
- [97] Wikipedia. Part-of-speech tagging. [https://en.wikipedia.org/wiki/Part-of-speech\\_tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging), February 2022.
- [98] Wikipedia. Rumiyah (magazine). [https://en.wikipedia.org/wiki/Rumiyah\\_\(magazine\)](https://en.wikipedia.org/wiki/Rumiyah_(magazine)), April 2022.
- [99] Wikipedia. Salafi jihadism. [https://en.wikipedia.org/wiki/Salafi\\_jihadism](https://en.wikipedia.org/wiki/Salafi_jihadism), February 2022. Published on Wikipedia.
- [100] Wikipedia. Supervised learning - applications. [https://en.wikipedia.org/wiki/Supervised\\_learning#Applications](https://en.wikipedia.org/wiki/Supervised_learning#Applications), February 2022.
- [101] Wikipedia. Supervised learning - applications. [https://en.wikipedia.org/wiki/Unsupervised\\_learning](https://en.wikipedia.org/wiki/Unsupervised_learning), February 2022.
- [102] Wikipedia. Support vector machines (svms) - applications. [https://en.wikipedia.org/wiki/Support-vector\\_machine#Applications](https://en.wikipedia.org/wiki/Support-vector_machine#Applications), March 2022.
- [103] Wikipedia. Timeline of al-qaeda attacks. [https://en.wikipedia.org/wiki/Timeline\\_of\\_al-Qaeda\\_attacks](https://en.wikipedia.org/wiki/Timeline_of_al-Qaeda_attacks), February 2022.