NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Fake Content Detection in the Information Exponential Spreading Era

Fernando Henrique Gregório Paulos Ferreira

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

FAKE CONTENT DETECTION IN THE INFORMATION EXPONENTIAL SPREADING ERA

by

Fernando Ferreira

Final report for dissertation as requirement for obtaining the master's degree in Information Management, with a specialization in Information Systems and Technologies Management.

Advisor: Roberto Henriques

November 2021

ABSTRACT

Recent years brought an information access democratization, allowing people to access a huge amount of information and the ability to share it, in a way that it can easily reach millions of people in a very short time. This allows to have right and wrong uses of this capabilities, that in some cases can be used to spread malicious content to achieve some sort of goal. Several studies have been made regarding text mining and sentiment analysis, aiming to spot fake information and avoid misinformation spreading. The trustworthiness and veracity of the information that is accessible to people is getting increasingly important, and in some cases critical, and can be seen has a huge challenge for the current digital era. This problem might be addressed with the help of science and technology. One question that we can do to ourselves is: How do we guarantee that there is a correct use of information, and that people can trust in the veracity of it? Using mathematics and statistics, combined with machine learning classification and predictive algorithms, using the current computation power of information systems, can help minimize the problem, or at least spot the potential fake information. One suggests developing a research work that aims to reach a model for the prediction of a given text content is trustworthy. The results were promising reaching a predicting model with good performance.

KEYWORDS

Data mining; Sentiment Analysis; Machine Learning; Fake Content.

INDEX

Table of Figures

1. INTRODUCTION

Currently we live in a time of online content exponential spreading and sharing. This brings enormous advantages for all mankind, which will perhaps be the time in which more people have access to more information throughout history. It also allows people having the capacity to use and spread wrong contents and information with the goal of disinform, promote violence or simply spread wrong information just for fun.

This great power that now almost everyone on the planet has in its hands, every time and everywhere, comes with a higher responsibility not only to people, but also for governments and institutions, that should think of better ways for coping this challenge than simple legislation and rules, that technically cannot stop the information spreading phenomena. What we can see nowadays is that technology has evolved too fast, and the governments and institutions that rule the countries, and the world, were left behind, without knowing what to do, or if they know what to do, they are too slow to do something with it. As a result, we have the big technology companies (or the people and other entities who own them) has the gatekeeper of this problematic phenomena, with all the risks and issues that this can have.

One of the best examples of this problematic were the 2020 presidential election in the United States, that had millions of fake content sharing using the social media technological platforms (Euronews 2021), and on the other hand, when things got violent, these companies banned some users and contents. What this shows is that we do not have the governments and institutions in control, but we have these companies doing it, using their power to censor information, with a perfectly reasonable explanation for doing it in this specific case, but it is known from historical facts that this kind of approach is a common practice in non-democratic countries.

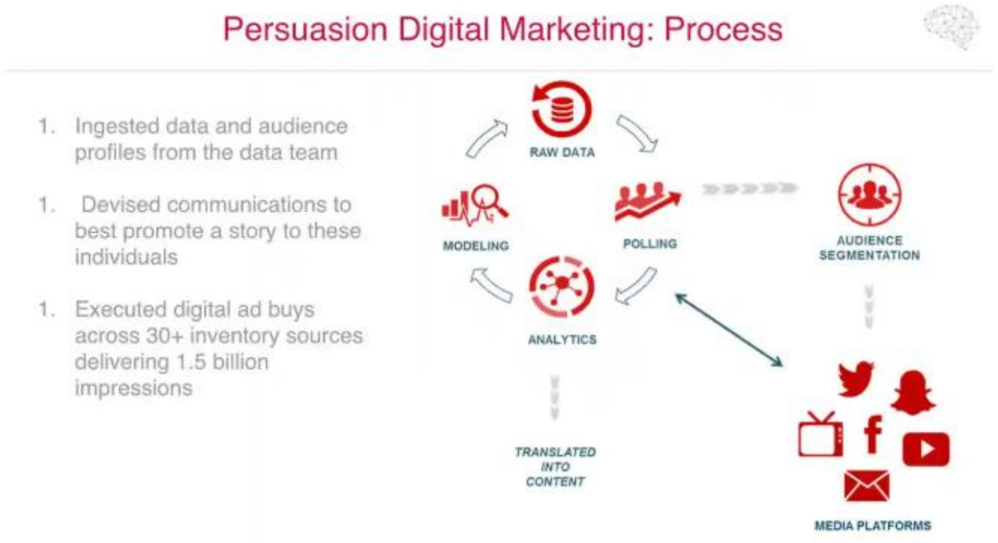
Science and technology can play a central role helping governments, regulators, and people in general, to address the challenge of fact checking, not with the objective of sensor it, but with the goal of spotting and warn people in a more automatic and fast way, trying to involve less human work, to avoid the inherent human bias associated and their limited capacity, as we currently have millions of contents being shared every single day.

The goal of this work is to study and investigate what is being done in industry and academic areas related with the topic of veracity information validation in online platforms and to propose an approach to spot fake information, like news or some text content, using predictive data mining methods for the classification, with various types of data analysis, modeling, and prediction, using an information source (Polígrafo n.d.) from Portugal.

1.1. BACKGROUND

The fake content detection is not a new challenge that came out recently, but it is getting more critical, once we have, for example, changes in countries' governments due to the use of advanced techniques of data manipulation and people segmentation, with the goal of influence their decisions. In many cases the information to a target person can have wrong contents that are used to influence the decision of that person in a certain direction. All these operations are being made without any type of regulation or control, and on the other hand generating massive profits to companies that own the technological platforms.

That was what many news networks content told us that happened with the Cambridge Analytica scandal, that has become public after the President of the United States 2016 election (Lewis Paul and Hilder Paul 2018). In the article published by The Guardian newspaper there are explained the techniques used by Trump campaign for micro-targeting US voters, using intensive survey research, data modelling and performance-optimizing algorithms, to target 10,000 different ads to different audiences in the months leading up to the election. The ads were viewed millions of times and this approach explores one human weakness, that is believing in something that they see a lot of times, even that might not be true (Li et al. 2014), people start believing. Below we have an example of a digital persuasion process that was used:



Persuasion Digital Marketing: Process

1. Ingested data and audience profiles from the data team

1. Devised communications to best promote a story to these individuals

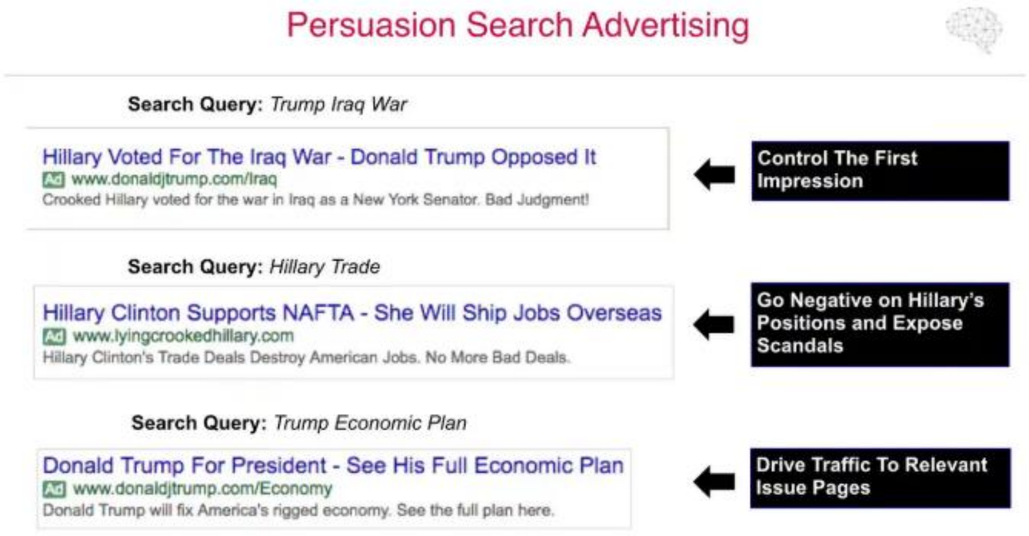
1. Executed digital ad buys across 30+ inventory sources delivering 1.5 billion impressions

Photograph: Cambridge Analytica

Figure 1 – Persuasion Digital Marketing Process (Li et al. 2014)

One might consider that these approaches might not be legal, but it is important to refer that none of the techniques used were considered illegal, and that they just used the available data and the tools from the social media platforms, like for Twitter quote: “Cambridge Analytica and the Trump campaign also used a new advertising technique offered by Twitter, launched at the start of the election year, which enabled clients to kickstart viral tweets.” (Lewis Paul and Hilder Paul 2018).

Google also gave some help to the Trump campaign were the company ensured that voters searching the words “Trump Iraq War” would encounter paid-for search results that were favorable to his campaign like the example below:



Persuasion Search Advertising

Search Query: *Trump Iraq War*

Hillary Voted For The Iraq War - Donald Trump Opposed It
Ad www.donaldjtrump.com/Iraq
Crooked Hillary voted for the war in Iraq as a New York Senator. Bad Judgment!

Control The First Impression

Search Query: *Hillary Trade*

Hillary Clinton Supports NAFTA - She Will Ship Jobs Overseas
Ad www.lyingcrookedhillary.com
Hillary Clinton's Trade Deals Destroy American Jobs. No More Bad Deals.

Go Negative on Hillary's Positions and Expose Scandals

Search Query: *Trump Economic Plan*

Donald Trump For President - See His Full Economic Plan
Ad www.donaldjtrump.com/Economy
Donald Trump will fix America's rigged economy. See the full plan here.

Drive Traffic To Relevant Issue Pages

Photograph: Cambridge Analytica

Figure 2 – Persuasion Search Advertising (Li et al. 2014)

Another example of this problematic is the recent COVID-19 pandemic situation, where after almost two years, we are still facing several challenges regarding the trustworthiness of information (Hankey 2020), and people’s privacy. In this article from Project Syndicate website, it was discussed the data privacy issue, arguing about why it is possible to develop algorithms that target the right people with the right information. The problematic is that everyone is giving their data to these companies, without almost no regulation, and these companies can take a huge profit from it, selling the information to other companies and governments. In the article is explained the case of South Korea where they link mobile-phone location data with individual travel histories, health data, footage from police- operated CCTV cameras, data from dozens of credit-card companies, and many other sources of information. This means that these companies have a huge number of individuals private information, that can be used not only for commercial purposes but also for surveillance and espionage. One might say that people agreed to share their personal information when they downloaded the apps, but people are surely not aware of all the uses that their information can have.

It is also important to understand why people seem so easy to manipulate. There are several papers and articles that study human psychology of why we are so vulnerable to the fake content and misinformation. We can find an interesting article that describes the main phycological situations where that can happen (Shane 2020), like for example the cognitive miserliness. In this case, people prefer simple and easier ways of understanding things, and don’t want to use mental effort to understand if something makes sense or try to validate through other sources, if some information is true or false. Other examples are cognitive dissonance or confirmation bias, that are related in the sense the people don’t like to see information that contradicts its beliefs. Many times, people prefer the information that confirms their ideas. Just for reference, as this is not the main subject of research, we can have the following misinformation causes, among others:

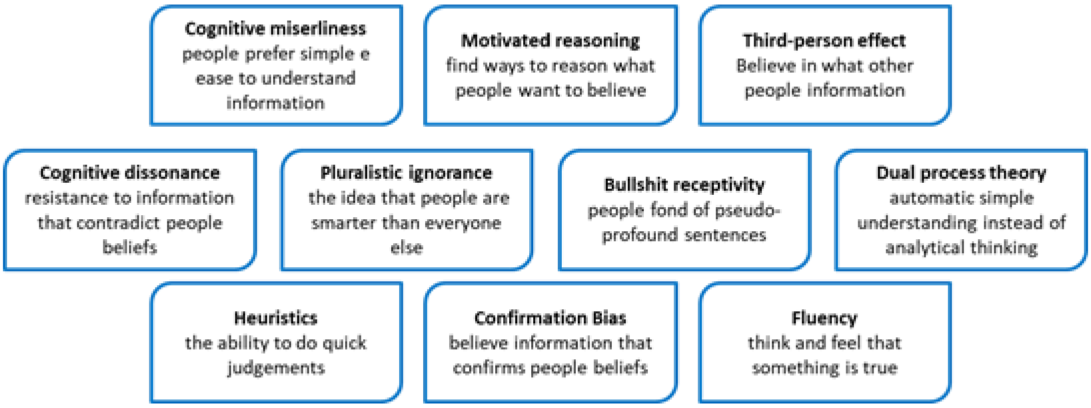
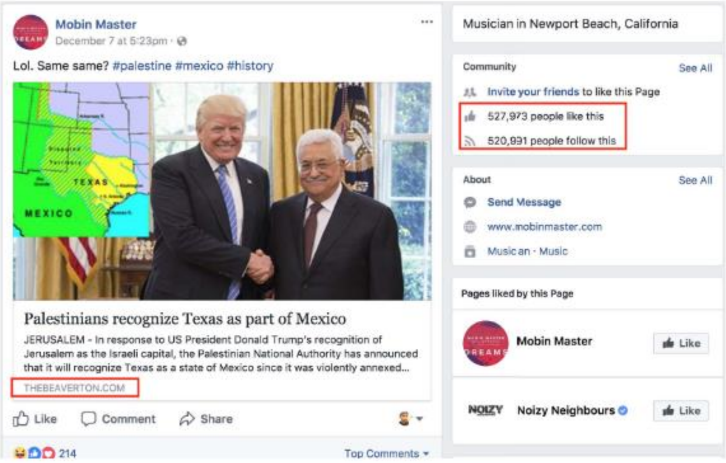

Cognitive miserliness
people prefer simple e ease to understand information
Motivated reasoning
find ways to reason what people want to believe
Third-person effect
Believe in what other people information
Cognitive dissonance
resistance to information that contradict people beliefs
Pluralistic ignorance
the idea that people are smarter than everyone else
Bullshit receptivity
people fond of pseudo-profound sentences
Dual process theory
automatic simple understanding instead of analytical thinking
Heuristics
the ability to do quick judgements
Confirmation Bias
believe information that confirms people beliefs
Fluency
think and feel that something is true

Figure 3 - Psychological misinformation causes

These techniques and situations explore the behavior of people when facing some news content and how combining compelling titles and headlines, with images and content, they are very easy to believe in the content. A study (Luz Yolanda Toro Suarez 2015) shows that 70% of Facebook users only read the headline of science stories before commenting or sharing, and this is one strong reason why we have so much fake content being spread. The example below, from the article "Media-Rich Fake News Detection: A Survey" (Parikh and Atrey 2018) shows a fake Facebook content published, that had hundreds of thousand likes and shares.



Mobin Master
December 7 at 5:23pm ·

Lol. Same same? #palestine #mexico #history

Palestinians recognize Texas as part of Mexico

JERUSALEM - In response to US President Donald Trump's recognition of Jerusalem as the Israeli capital, the Palestinian National Authority has announced that it will recognize Texas as a state of Mexico since it was violently annexed...

THEBEAVERTON.COM

Like Comment Share

214 Top Comments

Musician in Newport Beach, California

Community See All

Invite your friends to like this Page

527,973 people like this

520,931 people follow this

About See All

Send Message

www.mobinmaster.com

Music an · Music

Pages liked by this Page

Mobin Master Like

NOIZY Noizy Neighbours Like

Figure 1: An illustration of how the story titled "Palestinians recognizes Texas as part of Mexico" appears on Facebook [Source: http://www.facebook.com/]

Figure 4 – Example of misinformation content (Parikh and Atrey 2018)

In this content everything is prepared to exploit the human psychological weaknesses and bias, namely those people that are not fond of doing fact checking, and consequently believe it because it has an appealing headline with a very credible picture of two presidents shaking hands with a fake image behind.

The article (Li and Wu 2010), about text mining, studied the behavior of people in online forums and developed an algorithm for people segmentation using text mining and sentiment analysis. In this article the sentiment analysis and polarity were performed using k-means, that is a method that has the goal of partitioning n observations into k clusters, in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid). They also used support vector machines, that is a supervised learning model for classification, where it constructs a hyperplane, or set of hyperplanes, in a high- or infinite-dimensional space, where a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier. They used a methodology that started by the data preparation, with the downloading and crawling web pages, followed by data cleansing to remove inconsistency and noise, and data statistics analysis. After that they made the feature's extraction using sentiment analysis and clustering using K-means used for unsupervised prediction. Finally, they used classification using Support Vector Machine for forecasting.

The result of this work was the development of an algorithm to automatically analyze the emotional polarity of a texts. This algorithm combined with K-means clustering and SVM classification to develop integrated approach for online sports forums cluster analysis, using unsupervised clustering algorithm to group the forums into various clusters, with the center of each cluster representing a hotspot forum within the current time span. They conducted a forecast for the next time window. Empirical studies present strong proof of the existence of correlations between post text sentiment and hotspot distribution. Computation indicates both SVM and K-means produce consistent natural groupings results.

Another topic considered in my work was the fake online reviews for tourism business websites, were I found the work "The importance of behavioral data to identify online fake reviews for tourism businesses: A systematic review" from (Reyes-Menendez, Saura, and Filipe 2019), that analyzed the research work that has been done using the keywords "fake reviews" and "tourism".

This study of studies allowed to understand some patterns in terms of data and methodologies, where some perform analysis profiles of users who write reviews, where they seek patterns that can help better identify profiles that are more likely to generate false reviews. For other studies, the major unit of analysis is the content of online reviews, where they focus on two types of content, namely the textual content of reviews, on their linguistic aspects, such as the ratio of nouns to verbs, the type of words or the attributes used to write false reviews, and on detecting behavioral and emotional characteristics of users who write false reviews. This study is particularly relevant for my work, once I need to understand the best approach for features extraction and prediction of fake content.

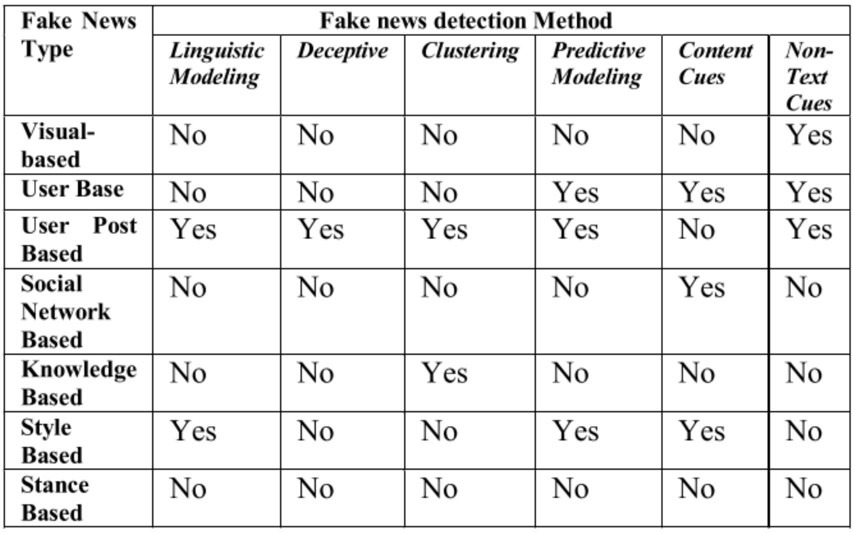
Another study analyzed, related with online fake reviews, was "Using supervised learning to classify authentic and fake online reviews" (Banerjee, Chua, and Kim 2015), where there were used supervised learning methods like Logistic Regression, Decision Trees, Neural Networks, Naïve Bayes, Random Forest or Support Vector Machines to classify the fake online reviews. In this study the dataset was relatively small with 1.800 reviews, and they have reached to some key findings related with the main features that influence models' capacity to spot fake reviews, namely the level of details, writing style and cognition indicators. They also found out that other studies using much more data and heavy algorithms can have a very good performance but are computationally intensive and might not be the best approach to near real time fake content spotting.

An interesting finding of the previous study is that using the reviews titles as a feature produces similar results when using the review description, being this an approach to explore when we know that we might not have access to large computational capacity or data.

Another study related with fake reviews analysis with the title "Towards automatic filtering of fake reviews" (Cardoso, Silva, and Almeida 2018), where they gave some examples on how this problem can impact companies and people's lives, like for example, a chef published fake negative reviews about rival restaurants on TripAdvisor, getting fired after his boss discovered the fraud on social media (Tylor 2015), or in another case, Samsung was fined for hiring spammers to post negative fake reviews about HTC smartphones (Chang 2010). They also made a comparison between scenarios considering online and offline learning, because static models created by offline learning methods may not be appropriated for spam review detection in real-world scenarios, considering that the characteristics of the reviews may change over time and the time-ordered nature of the reviews can be very important. In this context, it is better to use online learning methods, since the examples can be presented one at a time and there is no need to store all the examples in memory during the learning process. Therefore, online learning classifiers are appropriate for dynamic scenarios and, moreover, they are also indicated to deal with large-scale problems. This is particularly interesting for the scope of my work because many of the contents that will be presented to the models will be completely new and therefore an online learning approach it should be more appropriate.

In the previous work they combined sorted and non-sorted datasets by date, evaluating how the changes in the characteristics of reviews over time can influence the model's performance. For this matter it was also evaluated how other scenarios could influence model's performance like the polarity of the reviews (compliments vs complaints), or the use of real-world vs artificial reviews to train and evaluate the classifiers, or processing reviews of various type of services or products at the same time. They concluded that the performance is lower when using online learning and is higher when the data is presented ordered by date. They also found out that the performance is affected by the sentiment polarity of the reviews content. In my work I will also use this input information for the modeling of my problem and evaluate several scenarios to identify the more performant prediction model.

To understand the techniques needed for my current work I found a possible approach in the article "Fake News Detection Using Machine Learning approaches" (Manzoor, Singla, and Nikita 2019), where they described a set of techniques and methods used for social media posts depending on the type of content, that are shown in the following picture.



Fake News Type	Fake news detection Method					
	Linguistic Modeling	*Deceptive*	*Clustering*	*Predictive Modeling*	*Content Cues*	*Non-Text Cues*
Visual-based	No	No	No	No	No	Yes
User Base	No	No	No	Yes	Yes	Yes
User Post Based	Yes	Yes	Yes	Yes	No	Yes
Social Network Based	No	No	No	No	Yes	No
Knowledge Based	No	No	Yes	No	No	No
Style Based	Yes	No	No	Yes	Yes	No
Stance Based	No	No	No	No	No	No

Figure 5 – Fake news type vs Fake news methods (Manzoor et al. 2019)

Below we can find a brief explanation of the various types of fake contents from the table:

- Visual-based: uses graphical representation as content, this includes use of photoshopped images, video, and/or combination of both.
- User-based: is oriented towards certain audience by fake accounts and their target audience could represent certain age, gender, or culture groups.
- Post-based: are concentrated to be appeared on social media platforms. Post can be a Facebook post along with image or video and caption, a tweet, meme, among others.
- Network-based: are oriented towards certain members of a particular organization that also applied to group of friends on Facebook and group of mutually connected individuals on LinkedIn.
- Knowledge-based: contains scientific or reasonable explanation to an unresolved issue, these type of news stories are designed to spread false information.
- Style-based: focuses on the way of presenting to its readers, fake news is written by majority of people who are not journalists, the style of writing can be different.
- Stance-based: in-lines with above mentioned style-based type, stance is different in a sense that it focuses on how statements are being made in an article. Truthful news articles are written in a way to give sufficient information about the subject matter, and it is on readers to take way the meaning of the story.

1.2. Problem Identification

By the introduction and background written before one can argument that trustworthiness and veracity of the information that is accessible to people is becoming increasingly important and critical, because of the impacts that can have in people's lives, therefore can be seen has a big challenge that, with the existing scientific knowledge and information systems resources available can be addressed with the help of technology.

With all this fast evolution and constant change, technological mechanisms to control this information have been developed, restricting access, whether regarding true or false information, depending on the purpose of those who restrict it, as well as the targeted publication of right or wrong information, with the intention of influencing a target message recipient.

The USA 2020 Presidential election already mentioned in the document is a very good example of this where the President Donald Trump was blocked by the major social media platforms Twitter, Facebook, and Instagram, due to the publication of disputed content that was seen as inciting violence. The situation raises a new concern for governments and institutions that are responsible for ruling countries, and, in the end, they are responsible for ruling the world. This concern is the power that these big companies have, mostly technological based, but there are others with the same kind of power, to censor one of the pillars of modern democratic countries that is the free speech of their people. This concern is already being discussed and the fact checking community of IFCN in an article published (Tardáguila 2020) doesn't seem very happy with it, considering the censoring risk that comes with it, knowing that it is done by private companies, that are biased by their main goals that, in a capitalist organization of the world, is to maximize their profits.

The main problem is how do we create the awareness and confidence in people that some information that they are seeing is trustable and they can rely on it to make their judgements, opinions, and decisions. How can we create automatic analysis that help people to spot when some content is not true, even considering that people seem easy to manipulate and so vulnerable to the fake content and misinformation?

Answering to the following questions will contribute to the solution of the stated problem:

- How do different approaches in the methods and text features extraction can affect the veracity predictive of model's performance?
- How can technology provide help so that people can spot fake content in news websites or social media?
- Can fact checking activity be totally performed automatically by machines?

2. STUDY OBJECTIVES

2.1. Study Relevance

Currently we have many public and private entities, and people in general, concerned with this topic of content veracity, with news networks trying to make the information more reliable and trustfulness. The social media platforms also seem to be doing something, like inhibit certain types of information or classify information to allow recipients validate its veracity. There are several fact checking platforms all over the world like “Full Fact” from United Kingdom or “Poligrafo” from Portugal. The full list of signatories of this network can be found in IFCN (International Fact Checking Network) web site (The Poynter Institute 2021).

The social media platform Twitter seems to have initiatives to cope this challenge, as we all could witness in a recent controversy with the ex-President of the United States (Brian Fung 2020), regarding to an information shared that they associated a link for fact checking, meaning that the information shared was not trustable or might be wrong. A few weeks later they even banned Donald Trump from the platform, considering that he was inciting to violence, and this has been considered by many as an attack to one of the fundamental rights of democratic countries that is the free speech.

These companies, like Twitter or Facebook, have teams working on this topics, taking advantage of their access to information that others don’t have, using powerful platforms and infrastructures to automate this validations, that in some cases are still made by humans looking at images and reading texts and other contents, resulting in blocking them or restricting the access to it, like the hate speech control that is done by Facebook (Billy Perrigo 2020).

In the case of politically exposed and mediatic people, it is easier to spot because they have a special attention from this platforms and might have people dedicated just checking and following everything they post, but we can have someone not so mediatic posting content that is rapidly spread and when platforms try to stop it, it can be too late because it has been spread in many different formats and through many platforms. We can find some interesting insights in a study called “An Exploratory Study of COVID-19 Misinformation on Twitter” (Shahi, Dirkson, and Majchrzak 2020), that has some similarities with the research that the present work aims to do. In a very short way, they have gathered information from reliable sources of information, like the fact checkers from Poynter, crawling data from websites and then search Twitter for the ID’s that used the URL, that then were analyzed and categorized. They studied 92 fact checking websites and got their verdicts for the contents analyzed. Some of the results can be seen in the following picture (normalized) where it is very clear the amount of misinformation that currently is spread using Twitter platform:

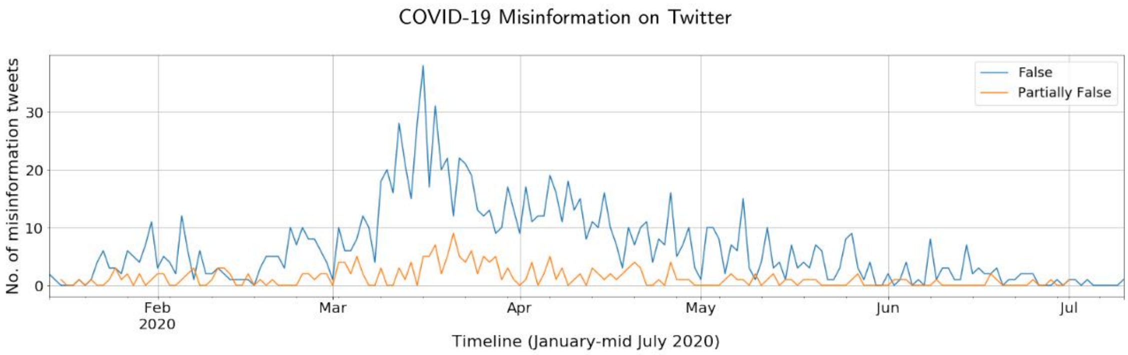


Figure 6 – Timeline of misinformation tweets (Shahi et al. 2020)

For further studies and investigations, a wider work is being done by researchers and can be extended in correlated topics like:

- Ability to develop a global identification number for contents, based on security and certification technology like blockchain, to subsequently guarantee the traceability and non-repudiation of certain information, and therefore the trustworthiness of its sources and the veracity of its content.
- Semantic analysis to classify the informative content of a given text to allow inferring if it is a fact or if it includes value judgments that influence the recipient in each political, doctrinal, or other sense.

2.2. Specific Objectives

This research work will consist in extracting data from a website, work on the data preprocessing and features extraction, before using classification text mining methods for data classification and model fitting, and finally analyze the model's performance considering the questions that we want to answer in this work. In the end there will be presented new information to these models, and it will evaluate the performance for a small subset of contents.

The model implementation will have as inputs the texts and the features extracted, manually or automated, and return a classification prediction. The technology used will be Python libraries like Beautiful Soup (Richardson 2016) for gathering and pre-processing data from web sites, NLTK (Klein 2006) for data processing and features extraction, Scikit-learn (Pedregosa et al. 2011) for models fitting and performance analysis.

3. METHODOLOGY

For this research work it will be used the following methodology:

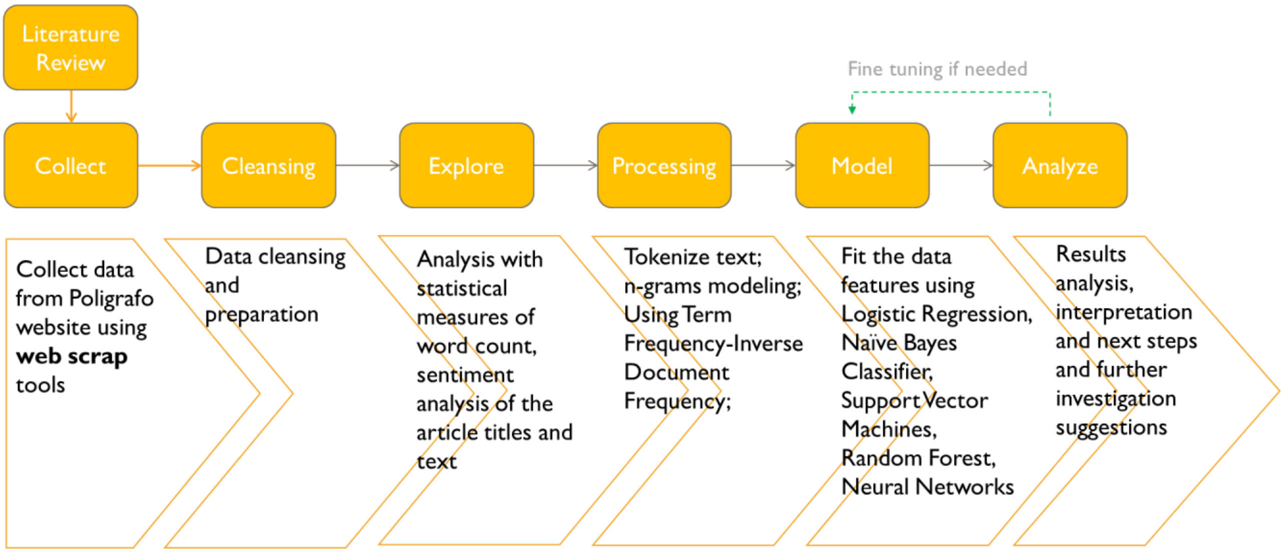
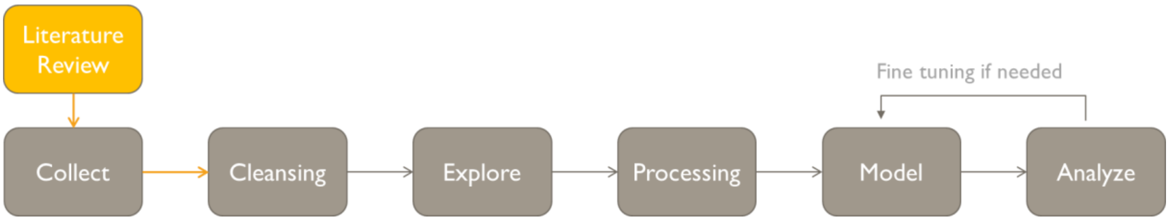

Literature Review
Fine tuning if needed
Collect
Cleansing
Explore
Processing
Model
Analyze
Collect data from Poligrafo website using **web scrap** tools
Data cleansing and preparation
Analysis with statistical measures of word count, sentiment analysis of the article titles and text
Tokenize text; n-grams modeling; Using Term Frequency-Inverse Document Frequency;
Fit the data features using Logistic Regression, Naïve Bayes Classifier, Support Vector Machines, Random Forest, Neural Networks
Results analysis, interpretation and next steps and further investigation suggestions

Figure 7 - Methodology diagram

The details of each step will be explained in the following sections.

3.1. Literature Review

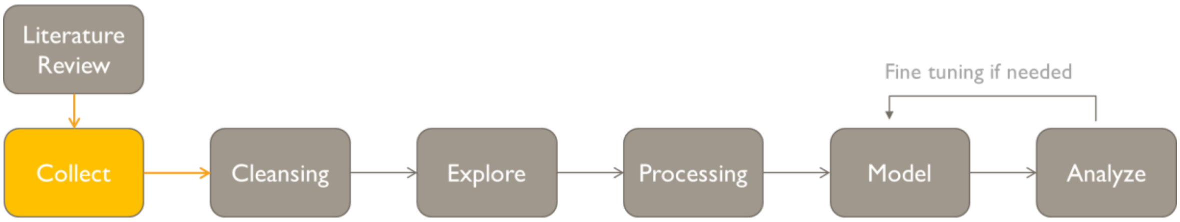


It was performed the **Literature Review** step, already stated in the previous sections of present document, to understand the problem, that the trustworthiness and veracity of the information that is accessible to people is becoming increasingly important and critical, because of the impacts that can have in people's lives, therefore can be seen has a big challenge, that with the existing scientific and systems resources available can be addressed with the help of technology.

Therefore, the goal of this work is to study and investigate what is being done in industry and academic areas related with the area of veracity information validation in online platforms and to propose an approach to spot fake information like news or some text content, using predictive data mining methods for the classification of texts, with various types of data analysis, modeling, and prediction, using various sources of information.

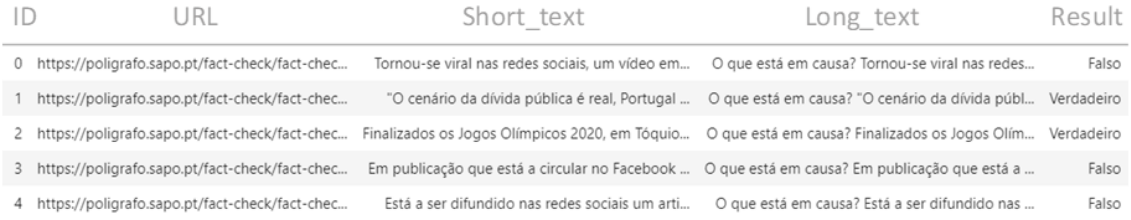
The main idea is to be able to help the human function of validating information, who have limited capacities for information that is available online, as well as reducing the bias associated with the interpretations and missuses of the information.

3.2. Collect



The **Collect** step started by data collection from the website poligrafo.pt (Polígrafo n.d.), that is a Portuguese online journalistic project, whose main objective is to verify the truth and not the lie, in the public space. It was used information from its "Fact-Checking" section and their classification if some content is true or false. Once they have a categorical classification with seven possible values, for the binary classification problem, some of them were filtered.

The data was extracted using Jupyter Notebook and Anaconda from the Portuguese poligrafo.pt (Polígrafo n.d.) web site and it contains four columns with the URL, a small text summary, a long text with all the explanation of the situation and the classification. Below we have a small subset of the data extracted.

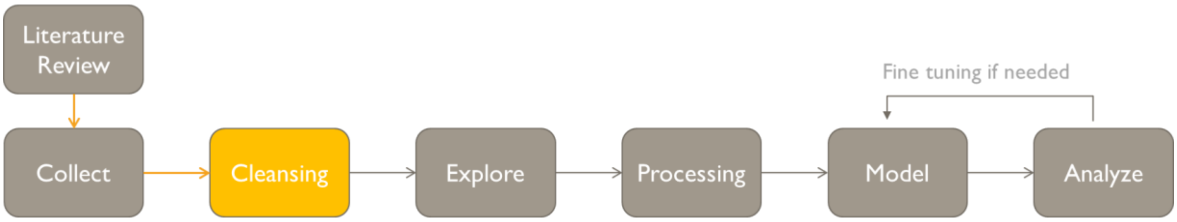


ID	URL	Short_text	Long_text	Result
0	https://poligrafo.sapo.pt/fact-check/fact-chec...	Tornou-se viral nas redes sociais, um vídeo em...	O que está em causa? Tornou-se viral nas redes...	Falso
1	https://poligrafo.sapo.pt/fact-check/fact-chec...	"O cenário da dívida pública é real, Portugal ...	O que está em causa? "O cenário da dívida públ...	Verdadeiro
2	https://poligrafo.sapo.pt/fact-check/fact-chec...	Finalizados os Jogos Olímpicos 2020, em Tóquio...	O que está em causa? Finalizados os Jogos Olím...	Verdadeiro
3	https://poligrafo.sapo.pt/fact-check/fact-chec...	Em publicação que está a circular no Facebook ...	O que está em causa? Em publicação que está a ...	Falso
4	https://poligrafo.sapo.pt/fact-check/fact-chec...	Está a ser difundido nas redes sociais um arti...	O que está em causa? Está a ser difundido nas ...	Falso

Figure 8 – Subset of data entries

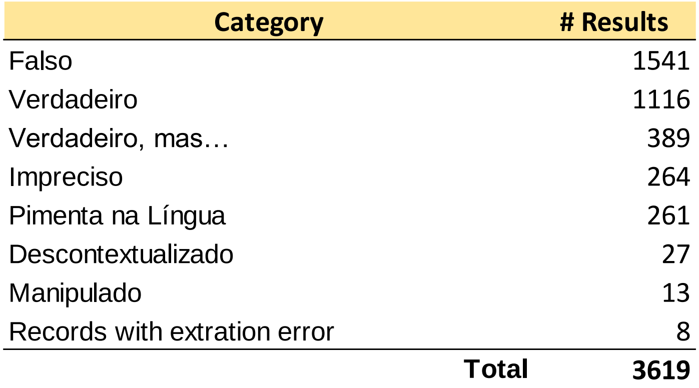
At the date of extraction (August 2021), it was possible to gather 3.619 records.

3.3. Cleansing



Data **Cleansing** step is important because we want to remove words and symbols that do not have any impact on the meaning of the sentences and texts. There were also removed words that do not have any impact on semantic meaning to the text that we call "stop words". There are several datasets of stop words for the English language like the NLTK framework, that is a platform for building Python programs to work with human language data.

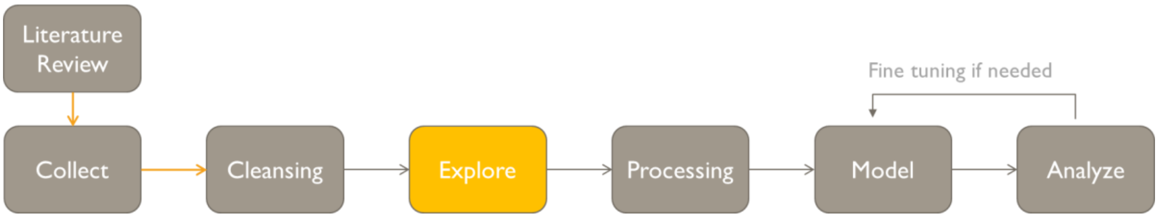
A simple analysis of the Result (target) variable also allowed to spot 8 records that needed to be excluded due to failure in the extraction process. We can also see that there is no need to take care of missing values, because all the records with no errors have a classification.



Category	# Results
Falso	1541
Verdadeiro	1116
Verdadeiro, mas…	389
Impreciso	264
Pimenta na Língua	261
Descontextualizado	27
Manipulado	13
Records with extration error	8
Total	3619

Figure 9 – Dataset classification counts per category

3.4. Explore



For the **Explore** step the data extracted didn't have features, therefore before starting the modeling process, it had to be transformed and extracted featured for the classification algorithms. After the first cleansing we have a dataset with 3611 classified and verified texts as shown in the figure below. The dataset is not balanced and we have 7 possible values for the taget variable.

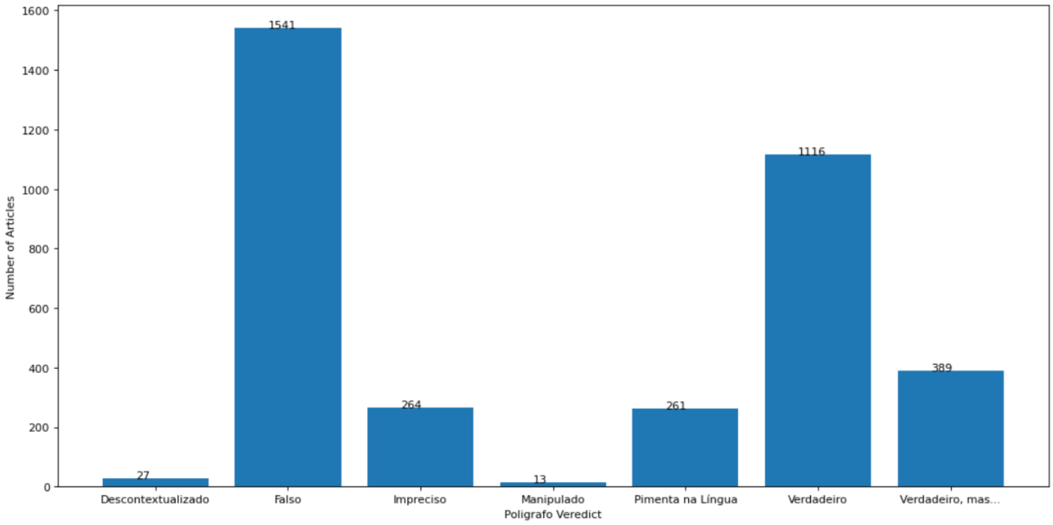
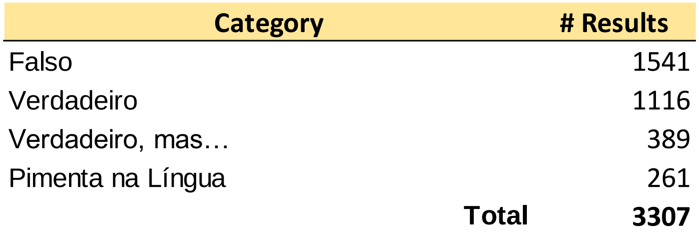


Figure 10 - Dataset classification histogram per category

Additionally for the binary classification problem that will be used in one of the modeling scenarios the dataset was filtered by the classification "Verdadeiro" and "Verdadeiro, mas…" as True (value=1) and the classifications "Falso" and "Pimenta na Lingua" as False (value=0).

We got a total of 3.307 records from the original dataset and have an almost balanced dataset.



Category	# Results
Falso	1541
Verdadeiro	1116
Verdadeiro, mas…	389
Pimenta na Língua	261
Total	**3307**

Figure 11 – Filtered dataset entries count

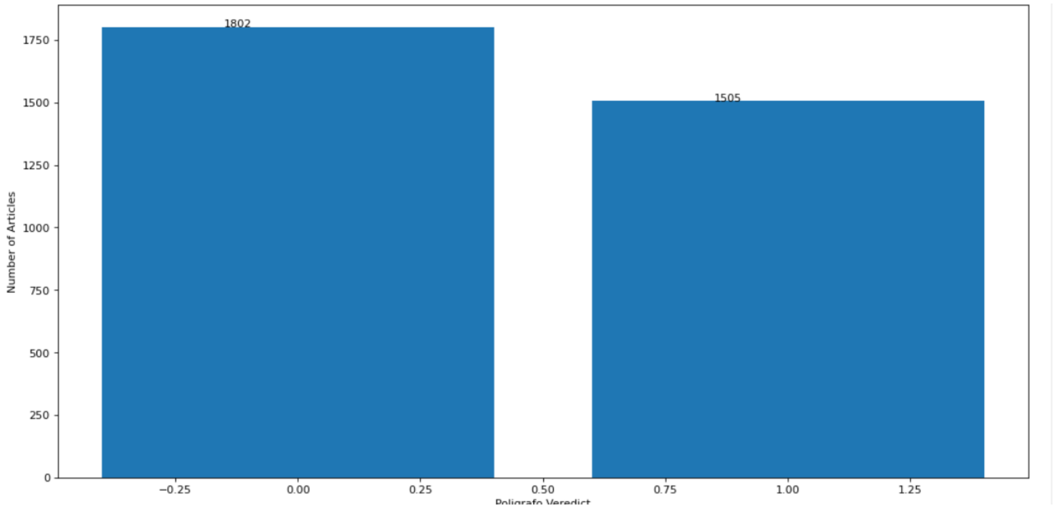


Figure 12 - Dataset classification histogram per binary category

For the binary problem we will consider 1902 False records and 1505 True records.

The following list of features were extracted from the texts, in order to understand the data and try to spot problems with the it:

1. Stopwords count – stopwords are words in the text that don't add much information and can be dropped for text analysis algorithms;

2. Punctuation count – The count of punctuation caracthers in the text;

3. Upper case count – The count of upper case words in the text;

4. Number count – The count of numbers in the text;

5. Characters count – count the number of caracthers of each text;

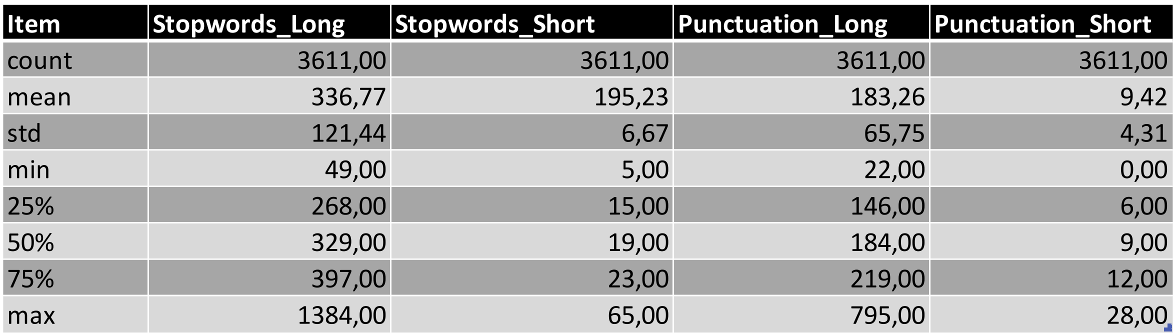
6. Word count – count the number of words of each text;

7. Average word – count the average number of words of each text;

8. Polarity – The sentiment polarity for a text is the orientation of the expressed sentiment, namely it determines if the text expresses the positive, negative or neutral sentiment based on a set of words considered positive or negative. Because the software libraries isn't prepared for portuguese language, the texts were first translated to english using the Google API before runing the polarity function;

9. Subjectivity – The sentiment subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Because the software libraries isn't prepared for portuguese language, the texts were first translated to english using the Google API, before runing the subjectivity function.

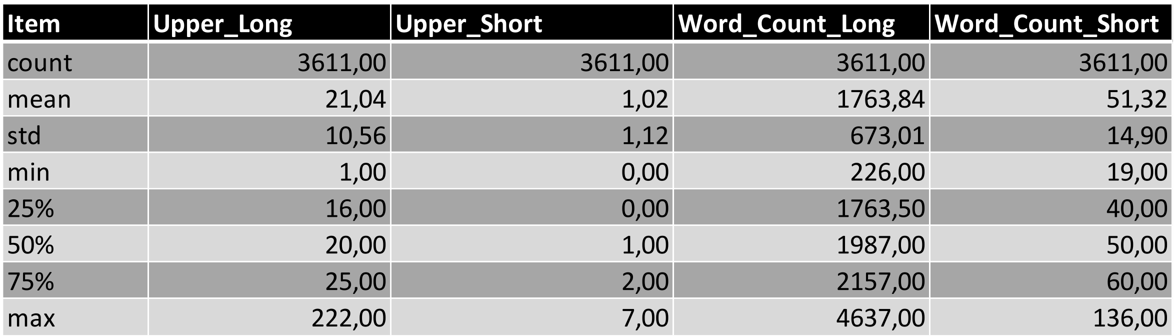
After extracting the features there was analyzed the statistical information to understand what variables could be used for modeling. Each of the counting features were extracted from the long text input variable and from the short text input variable. The following table shows the basic statistical information from the features extracted:



Item	Stopwords_Long	Stopwords_Short	Punctuation_Long	Punctuation_Short
count	3611,00	3611,00	3611,00	3611,00
mean	336,77	195,23	183,26	9,42
std	121,44	6,67	65,75	4,31
min	49,00	5,00	22,00	0,00
25%	268,00	15,00	146,00	6,00
50%	329,00	19,00	184,00	9,00
75%	397,00	23,00	219,00	12,00
max	1384,00	65,00	795,00	28,00

Figure 13 – Table with features Stopwords and Punctuation statistics

From the statistical analysis of the features in the table above we can see a low standard deviation for the feature "Stopwords_Short", that means that the data is mostly clustered around the mean. We can also spot that the features "Stopwords_Long" and "Punctuation_Long" have outliers since the maximum value is four times the mean. The outliers will be excluded from the dataset before the modeling process.



Item	Upper_Long	Upper_Short	Word_Count_Long	Word_Count_Short
count	3611,00	3611,00	3611,00	3611,00
mean	21,04	1,02	1763,84	51,32
std	10,56	1,12	673,01	14,90
min	1,00	0,00	226,00	19,00
25%	16,00	0,00	1763,50	40,00
50%	20,00	1,00	1987,00	50,00
75%	25,00	2,00	2157,00	60,00
max	222,00	7,00	4637,00	136,00

Figure 14 – Table with features Upper Case count and Word count statistics

The "Upper_Short" feature doesn't seem to add value explaining the data, once usually the news short description has less words than the long text description, and the phrase starts with one upper case letter, that is close to the mean (1,02). The "Upper_Case" feature has the same behavior of other features and seems to have outliers once the maximum value is very far from the mean.

Item	Num_Count_Long	Num_Count_Short	Char_Count_Long	Char_Count_Short
count	3611,00	3611,00	3611,00	3611,00
mean	28,81	0,43	7235,42	317,71
std	13,96	0,78	2565,62	89,97
min	0,00	0,00	1012,00	111,00
25%	30,00	0,00	6420,50	252,00
50%	33,00	0,00	7603,00	309,00
75%	36,00	1,00	8629,00	369,00
max	88,00	8,00	24423,00	801,00

Figure 15 – Table with features Number count and Character count statistics

We can see that we usually have very few numbers in the short texts ("Num_Count_Short") that were extracted, meaning that this variable will be discarded from the modeling process. The character count ("Char_Count_Long" and "Char_Count_Short" features), as expected, has the same behavior of the word count features and are highly correlated, therefore should be excluded from the modeling process.

Item	Avg_Word_Long	Avg_Word_Short	Polarity_Long	Polarity_Short
count	3611,00	3611,00	3611,00	3611,00
mean	5,33	5,24	0,06	0,07
std	0,16	0,41	0,05	0,15
min	4,50	3,80	-0,18	-1,00
25%	5,20	5,00	0,03	0,00
50%	5,30	5,20	0,06	0,05
75%	5,40	5,50	0,09	0,14
max	6,40	6,90	0,40	1,00

Figure 16 – Table with features Average word and Polarity statistics

The average words count ("Avg_Word_Long" and "Char_Count_Short") have a low standard deviation as expected once they were averaged, therefore the values are all clustered around the mean. The polarity features ("Polarity_Long" and "Polarity_Short") are both clustered around the mean but with outliers.

Item	Subjectivity_Long	Subjectivity_Short
count	3611,00	3611,00
mean	0,42	0,34
std	0,07	0,19
min	0,14	0,00
25%	0,38	0,21
50%	0,43	0,33
75%	0,47	0,46
max	0,61	1,00

Figure 17 – Table with feature Subjectivity statistics

We can see that when using the short text ("Subjectivity_Short") to calculate the subjectivity feature, the standard deviation is very high, meaning that this measure is spread out and that might not have enough information for the calculation feasibility. We can also conclude that this calculation returns better results using the long text ("Subjectivity_Long"), because it has a lower standard deviation, and the values never reach the full range between 0 and 1.

The sentiment analysis information was calculated using the python TextBlob library from the NLTK(Klein 2006) framework. Calculating the sentiment of a text through TextBlob provides numeric values for polarity and subjectivity. The numeric value for polarity describes how much a text is negative or positive. Subjectivity describes how much a text is objective or subjective. TextBlob uses a lexicons file with the word's classification. The file does not contain stopwords, because they do not have any sentiment. Each word is defined in the lexicon file with their part of speech (POS), polarity, subjectivity, intensity, and confidence. When calculating the sentiment for a single word, TextBlob uses the "averaging" technique that is applied on values of polarity to compute a polarity score for a single word. A similar operation applies to every single word, and we get a combined polarity for longer texts.

The figure below allows to identify that usually the true texts have a very positive polarity score (Negative←→Positive) and a high subjectivity score (Facts←→Opinions). In the figure below we have used the long text variable.

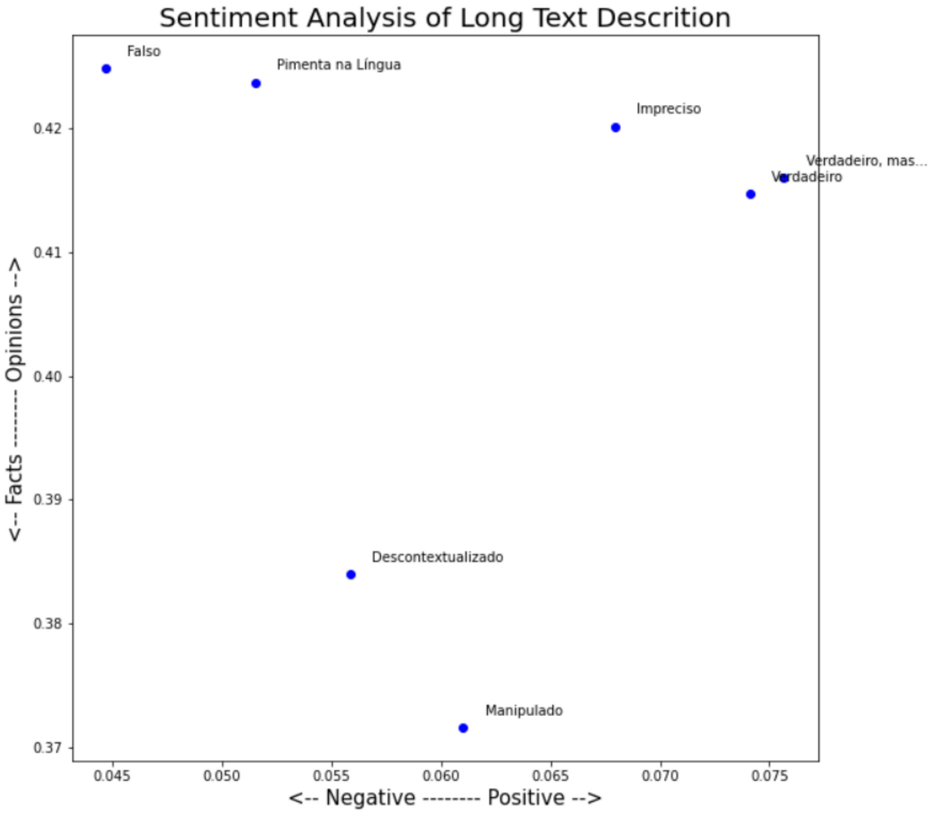


Figure 18 – Sentiment Analysis long text variable

When looking at the short text variable for the True labels it has also the same behavior.

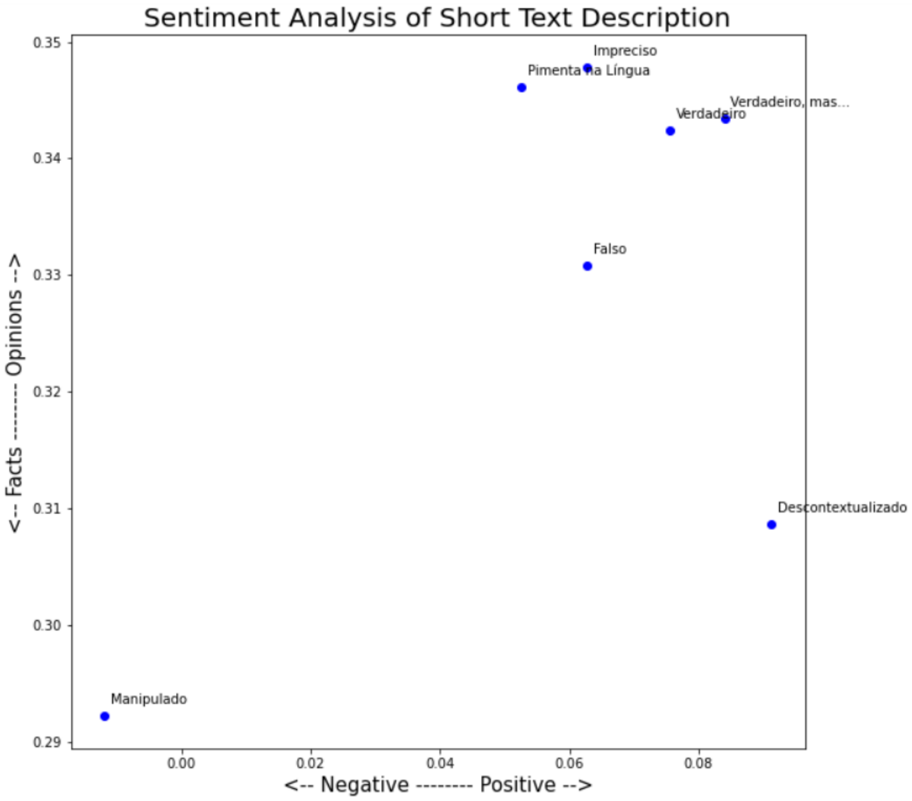


Figure 19 – Sentiment Analysis short text variable

We can also see that both polarity and subjectivity have a normal distribution for the long text variable.

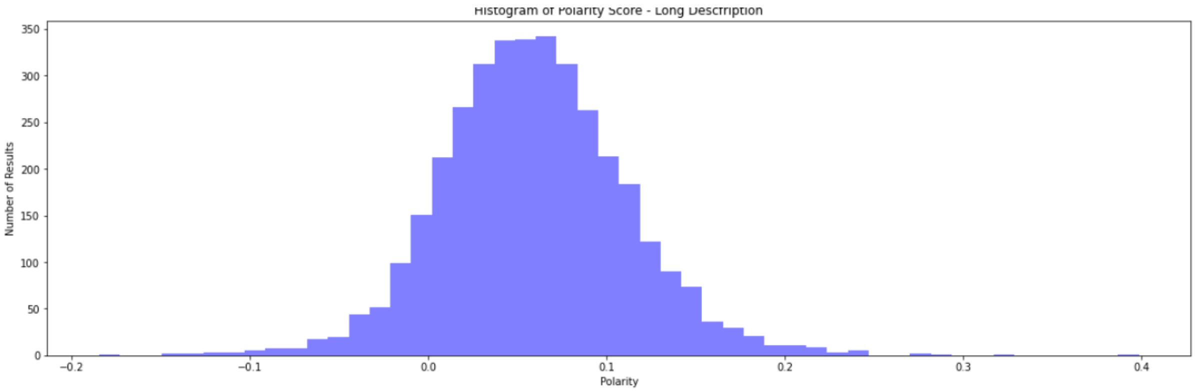


Figure 20 – Polarity score histogram long text

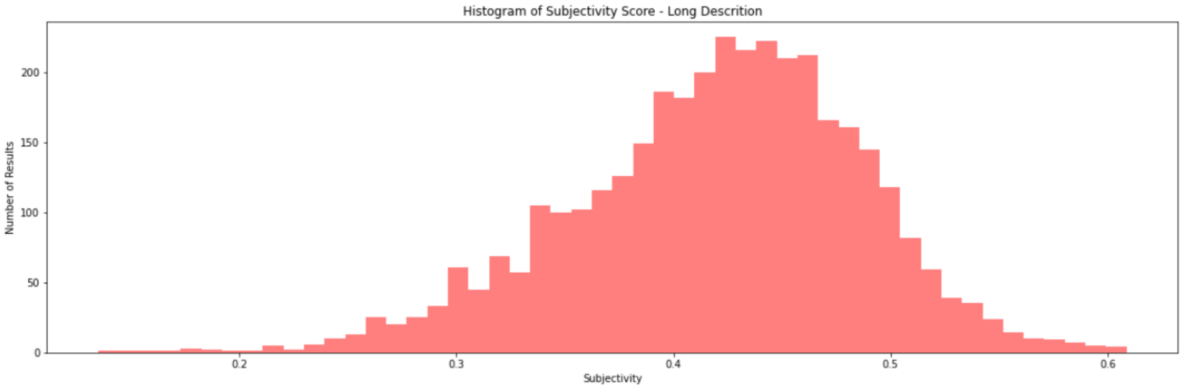


Figure 21 – Subjectivity score histogram long text

Although, for the short text variable the sentiment analysis doesn't have a normal distribution.

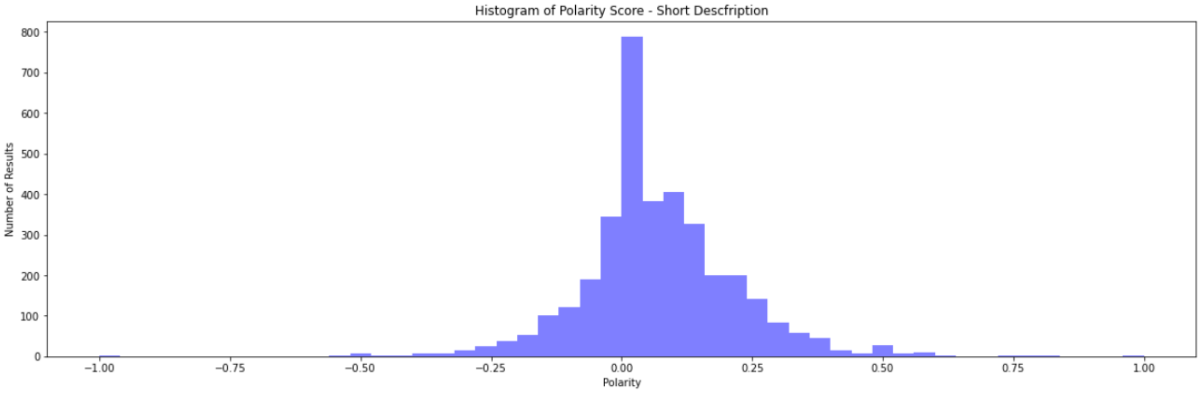


Figure 22 - Polarity score histogram short text

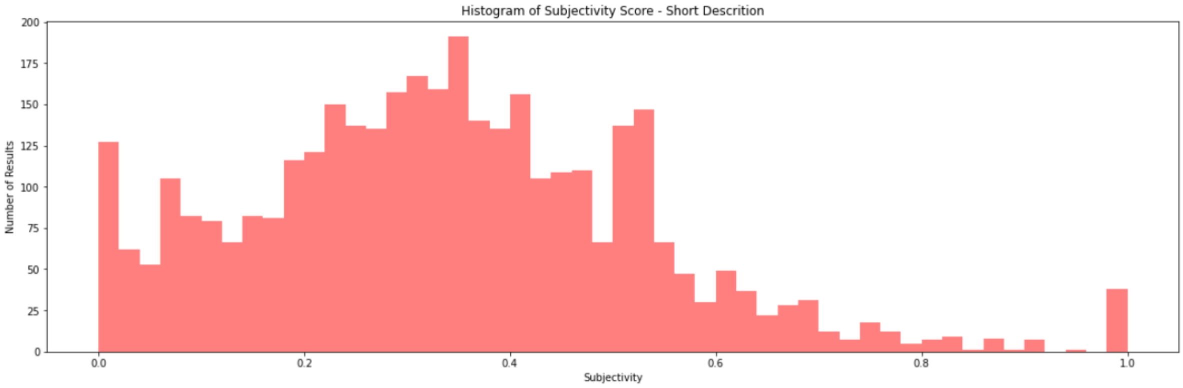
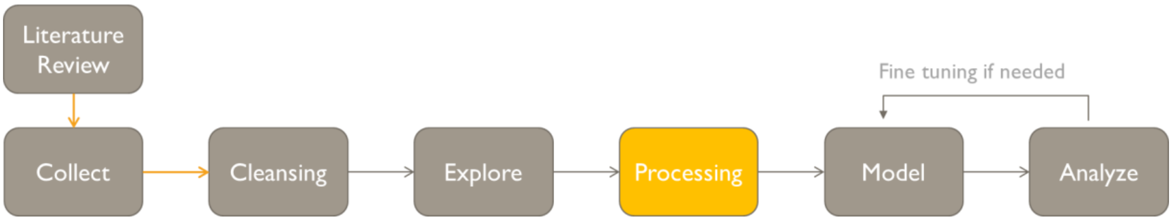


Figure 23 - Subjectivity score histogram long text

3.5. PROCESSING



In the **Processing** step it is important to state that the work will be focused only on text content, not considering the images, nor the social media user's data, nor the networks used to spread the information. There are also several methods and approaches for the fake content detection, namely:

- **Linguistic Features based Methods** – The features extraction can be based on Ngrams extraction using TFIDF (Term Frequency Inverse Document Frequency), punctuation that can help the fake news detection algorithm to differentiate between deceptive and truthful texts, psycho-linguistic features in order to extract psycho- linguistic features like word count and emotional tone extraction, readability that includes extraction of content features such as the number of characters, complex words, long words, number of syllables, word types, and number of paragraphs. Having these content features allow to perform readability metrics, such as Flesch-Kincaid, Flesch Reading Ease, Gunning Fog, Automatic Readability Index (ARI) and syntax analysis based on CFG (Context-free grammar). From the methods above I will use the basic feature extraction related with word count, punctuation and TFIDF.

- **Deception Modeling based Methods** – Relies on theoretical approaches, namely Rhetorical Structure Theory (RST) and Vector Space Modeling (VSM). The RST procedural analysis captures the logic of a story in terms of functional relations among different meaningful text units and describes a hierarchical structure for each story. The VSM is used to identify rhetorical structure relations in RST resulted sets. VSM interprets every news text as vectors in high dimensional space, this requires the extracted text to be modeled in a suitable manner for the application of various computational algorithms. I will not use any of these methods in my work.

- **Clustering based Methods** – Clustering is a known method to compare a clustering package to help differentiate news reports based on their similarity based on chosen clustering algorithm. The k-nearest neighbor approach, clustering similar news reports based on the normalized frequency of relations. The ability of this model to detect the deceptive value of a new story is measured based on the principle of coordinate distances.

- **Predictive Modeling based Method** – There are several regressive models that can be applied like logistic regression.

- **Content Cues based Method** - This method leverages two different analyses, namely Lexical and Semantic Levels of Analysis, that is the choice of vocabulary plays an important role in convincing readers to believe in the story. Automated methods can be used to extract stylometric features of the text (i.e., part of speech, word length and subjective terms), that can be used to discriminate between two journalistic formats. Also, can have Syntactic and Pragmatic Levels of Analysis, that is a pragmatic function of headlines that invokes reference to forthcoming parts in the discourse. This is done by referring to forthcoming parts in the news story. Headlines are written to fill empty thoughts with leveraging ensuing text.

- **Non-Text Cues based Methods** – focused on the non-text content of the news story is highly valuable in terms of convincing its readers to believe in contaminated news. This method leverages two different analyses: Image Analysis: Strategic use of images is a known key method to manipulate emotion in observers, and User Behavior Analysis: User Behavior Analysis is content-independent method largely useful to assess how readers engage with news once they are lured into the story. News produces must drive traffic to their original site from multiple avenues, such as, click-ads, social media presence, promotions.

NLTK (Klein 2006) framework provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. These libraries have functions that allow to extract features, namely:

- **Classification** – Using document classification, namely Part-of-Speech Tagging, Exploiting Context, Sequence Classification, Sentence Segmentation, Identifying Dialogue Act Types, Recognizing Textual Entailment.

- **Tokenization** - A tokenizer that divides a string into substrings by splitting on the specified string.

- **Stemming** – That is the process of producing morphological variants of a root/base word. For example, a stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate”.

- **Lemmatizing** – That is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is like stemming but it brings context to the words. So, it links words with similar meaning to one word.

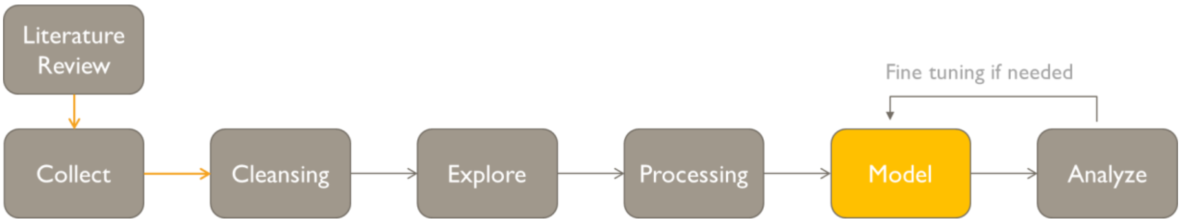
- **Parsing** – Used to derive syntax trees for sentences and to derive other kinds of tree structure, such as morphological trees and discourse structures.

- **TF-IDF** – Term frequency (TF) is how often a word appears in a document, divided by how many words there are. Term frequency is how common a word is, inverse document frequency (IDF) is how unique or rare a word is.

- **Bag of words** – In this case a given text, like a sentence or a document, is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity.

For the scope of the present work there were used only the TF-IDF feature extraction method and other syntax statistic measures and sentiment analysis already identified.

3.6. Model



The **Model** implementation step has as inputs the texts and several approaches and prediction analysis and scenarios. The technology used were the Python libraries for gathering and pre-processing data from the web site, for data processing and features extraction and for models fitting and performance analysis. There are several possible approaches for text mining analysis and in the present work there were made variations on those approaches and analyzed the results, so that it is possible to identify the strengths and weakness of each one.

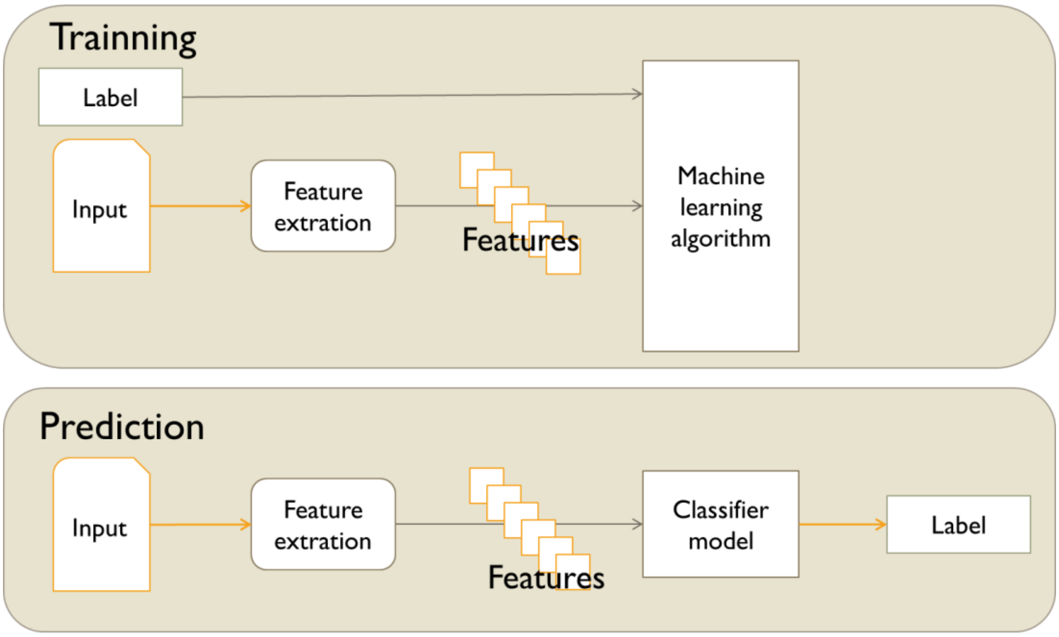


Figure 24 - Train and Test approach

The target variable is categorical, so it was decided to analyze the performance of the models considering the classification of all the existing values versus transforming the problem in a binary one, once the main goal of this work is to try to determine if a given text is true or false.

Another variation was considering a manual feature extraction and selections, versus counting feature extraction and compare the results of the several models. In the manual feature extraction, it will also be compared a variable selection approach versus using all variables extracted. The six possible scenarios can be seen in the following diagram:

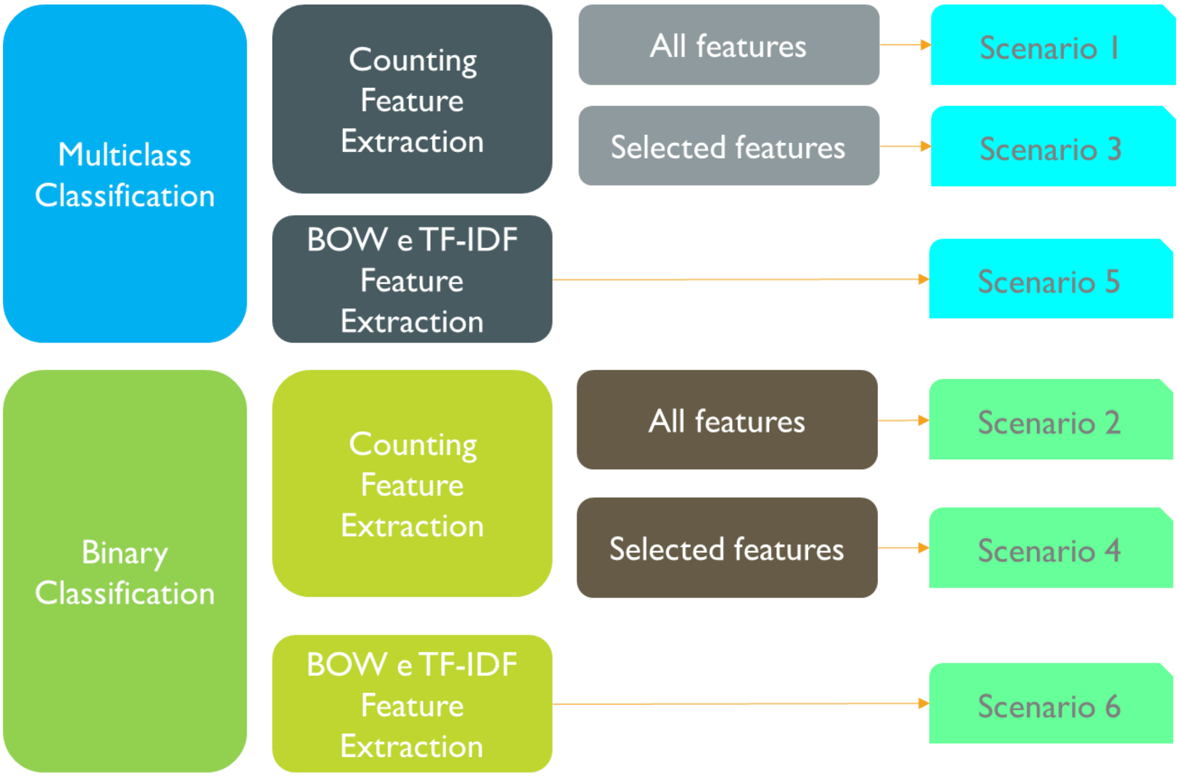

Multiclass Classification
Counting Feature Extraction
All features
Scenario I
Selected features
Scenario 3
BOW e TF-IDF Feature Extraction
Scenario 5
Binary Classification
Counting Feature Extraction
All features
Scenario 2
Selected features
Scenario 4
BOW e TF-IDF Feature Extraction
Scenario 6

Figure 25 – Scenario's mapping

The main steps followed for each scenario was:

1. Import necessary Python libraries and data from Poligrafo website.

2. Data pre-processing and cleansing, removing information not relevant for the modeling fitting and prediction. Text translation to use the sentiment analysis libraries.

3. Features extraction (word count, character count, average number of words, Punctuation count, upper case count, stop words count, numeric count, polarity, and subjectivity).

4. Statistical and correlation analysis where we found that we have correlation between variable higher than 0.8, that was dropped in iteration scenario for model comparison.

5. Model with the eight different algorithms, namely:

   ○ **Gaussian Naive Bayes** – In the Scikit Learn Python module (Pedregosa et al. 2011) the Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. The Gaussian Naive Bayes implements an algorithm for classification, considering that the likelihood of the features is assumed to be Gaussian, where the parameters σ_y and μ_y are estimated using maximum likelihood.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma^2_y}\right)$$

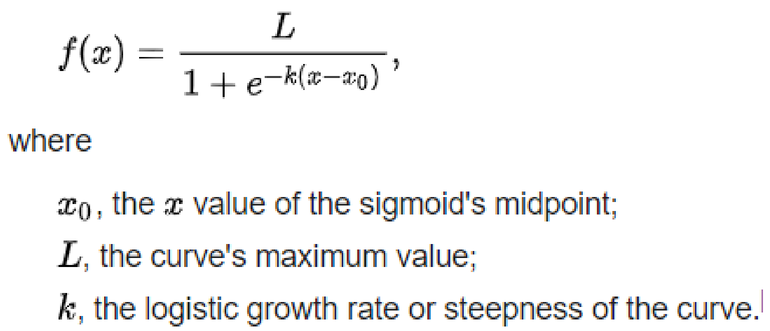
- **K Nearest Neighbors** (Pedregosa et al. 2011) – implements unsupervised nearest neighbors learning. It acts as a uniform interface to three different nearest neighbors' algorithms:

  - **Brute Force** – Fast computation of nearest neighbors is an active area of research in machine learning. The naivest neighbor search implementation involves the brute-force computation of distances between all pairs of points in the dataset: for N samples in D dimensions, this approach scales as O[DN2]. Efficient brute-force neighbors searches can be very competitive for small data samples. However, as the number of samples N grows, the brute-force approach quickly becomes infeasible.

  - **KD Tree** – To address the computational inefficiencies of the brute-force approach, a variety of tree-based data structures have been invented. In general, these structures attempt to reduce the required number of distance calculations by efficiently encoding aggregate distance information for the sample. The basic idea is that if point A is very distant from point B, and point B is very close to point C, then we know that points A and C are very distant, without having to explicitly calculate their distance. In this way, the computational cost of a nearest neighbor's search can be reduced (Bentley 1975). This is a significant improvement over brute-force for large datasets. One approach to taking advantage of this aggregate information was the KD tree data structure (short for K-dimensional tree), which generalizes two-dimensional Quad-trees and 3-dimensional Oct-trees to an arbitrary number of dimensions. The KD tree is a binary tree structure which recursively partitions the parameter space along the data axes, dividing it into nested orthotropic regions into which data points are filed. The construction of a KD tree is very fast: because partitioning is performed only along the data axes, no D-dimensional distances need to be computed. Once constructed, the nearest neighbor of a query point can be determined with less distance computations. Though the KD tree approach is very fast for low-dimensional (D<20) neighbors searches, it becomes inefficient as D grows very large: this is one manifestation of the so-called "curse of dimensionality".

  - **Ball Tree** – To address the inefficiencies of KD Trees in higher dimensions, the ball tree data structure was developed. Where KD trees partition data along Cartesian axes, ball trees partition data in a series of nesting hyper-spheres (Omohundro 1989). This makes tree construction more costly than that of the KD tree but results in a data structure which can be very efficient on highly structured data, even in very high dimensions. A ball tree recursively divides the data into nodes defined by a centroid **C** and radius *"r"*, such that each point in the node lies within the hyper-sphere defined by *"r"* and **C**. The number of candidate points for a neighbor search is reduced through use of the triangle inequality (|x+y|<=|x|+|y|). With this setup, a single distance calculation between a test point and the centroid is sufficient to determine a lower and upper bound on the distance to all points within the node. Because of the spherical geometry of the ball tree nodes, it can out-perform a KD-tree in high dimensions, though the actual performance is highly dependent on the structure of the training data.

- **Logistic Regression** (Wikipedia 2022) – despite its name, it is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. The probabilities describing the possible outcomes of a single trial are modeled using a logistic function. A logistic function or logistic curve is a common S-shaped curve (sigmoid curve) with equation:



$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}},$$

where

x_0, the x value of the sigmoid's midpoint;

L, the curve's maximum value;

k, the logistic growth rate or steepness of the curve.

- **Decision Trees** – are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model (Pedregosa et al. 2011).

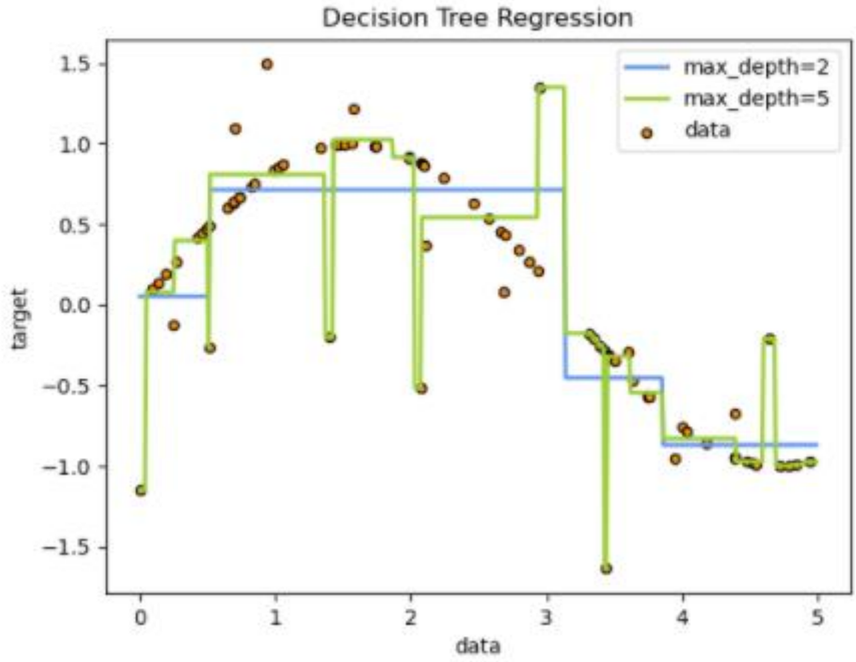


Figure 26 – Decision Tree Regression Example (Pedregosa et al. 2011)

○ **Support Vector Machines** (Wikipedia 2022) - are a set of supervised learning methods used for classification, regression, and outliers' detection. The simple Linear SVM can be explained as follows. Given a training dataset of n points of the form:

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n),$$

where the y_i are either 1 or −1, each indicating the class to which the p $\mathbf{x}_i$ t belongs. Each $\mathbf{x}_i$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\mathbf{x}_i$ for which y_i = 1 from the group of points for which y_i = -1, which is defined so that the distance between the hyperplane and the nearest point $\mathbf{x}_i$ from either group is maximized.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

  - If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

  - SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

- **Multinomial Naive Bayes** – In the Scikit Learn Python module (Pedregosa et al. 2011) the Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. The Multinomial Naive Bayes implements the naive Bayes algorithm for multinomially distributed data and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although TF-IDF vectors are also known to work well in practice).

- **Random Forest** – The Scikit-learn (Pedregosa et al. 2011) module includes two averaging algorithms based on randomized decision trees: the Random Forest algorithm and the Extra-Trees method. Both algorithms are perturb-and-combine techniques specifically designed for trees. This means a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers.

  In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size maximum number of features. The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit. The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors can cancel out. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant, hence yielding an overall better model.

- **Neural Networks** – Multi-layer Perceptron (MLP) from Scikit-learn (Pedregosa et al. 2011) is a supervised learning algorithm that learns a function by training on a dataset, where "*m*" is the number of dimensions for input and "*o*" is the number of dimensions for output. Given a set of features $X = x_1, x_2, \ldots, x_m$ and a target "*y*", it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure below shows a one hidden layer MLP with scalar output.

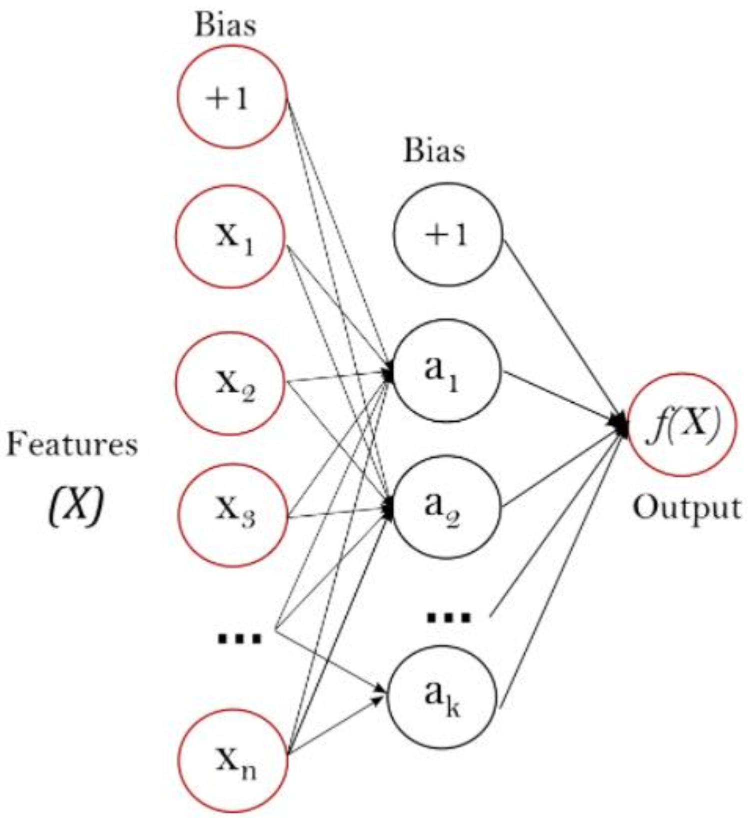


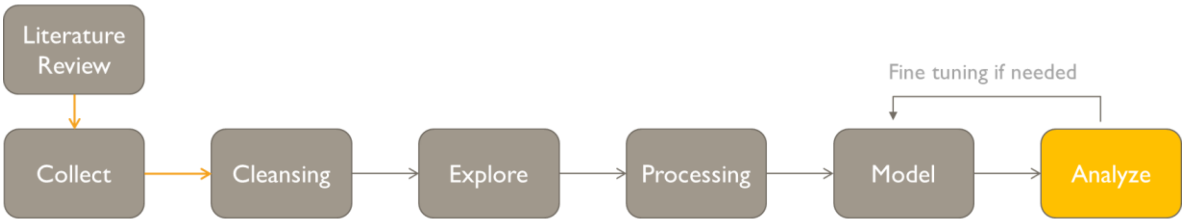
Figure 27 – One hidden layer MLP (Pedregosa et al. 2011)

6. Analyze and compare the eight models' performance in each scenario.
7. Perform fine tuning for the best three models of all scenarios

The first kind of evaluation to be made is if the model meets the research objectives and seek to determine if there is some reason why the model might be deficient. For all models there will be analyzed the following scores and metrics:

- **Classification Accuracy** – That is the ratio of number of correct predictions to the total number of input samples.
- **Classification Precision** - That is the ratio of number of correct positive results divided by the number of positive results predicted by the classifier
- **Confusion Matrix** – Matrix as output that describes the complete performance of the model. It will be evaluated the 4 variations: True Positives, True Negatives, False Positives, False Negatives.
- **F1 Score** – F1 Score is the Harmonic Mean between precision and recall:
  - **Precision** - The number of true positive results divided by the number of positive results predicted by the classifier.
  - **Recall** - The number of true positive results divided by the number of positive results in the sample data.

3.7. ANALYZE



The following sections present the several scenarios of the Analyze step.

3.7.1. Scenarios Analysis

In this section it will be presented the six modeling scenarios to spot the best approach and performant models.

8. **Scenario 1** – The first scenario was a multiclass classification problem, with no pre-processing of texts, counting features extraction and no filtering. In this case we found very poor performance for all models. The best ones were logistic regression and random forest, but in any case, **with bad scores** as we can see in the table below:

Model	Accuracy	Precision	Recall	F1 Score
Gaussian Naive Bayes	33%	36%	33%	32%
K Nearest Neighbors	39%	35%	39%	36%
Logistic Regression	44%	43%	53%	42%
Decision Trees	41%	33%	45%	35%
Support Vector Machine	32%	71%	43%	21%
Multinomial Naive Bayes	14%	37%	14%	13%
Random Forest	43%	38%	44%	36%
Neural Networks	42%	38%	57%	35%

Figure 28 – Scenario 1 scores

9. **Scenario 2** – The second scenario was very similar to the first one, with the difference that considered a binary classification, were the non-binary results were excluded, with no pre-processing, counting features extraction and no filtering. The performance was better than in first scenario, but **still very poor**. Nevertheless, it allows to conclude that in this kind of problem the models perform better having a binary classification problem than with multiple values in the target variable.

Model	Accuracy	Precision	Recall	F1 Score
Gaussian Naive Bayes	54%	53%	54%	53%
K Nearest **Neighbors**	54%	53%	54%	51%
Logistic Regression	59%	59%	59%	56%
Decision Trees	55%	54%	55%	54%
Support Vector Machine	50%	62%	50%	38%

Multinomial Naive Bayes	54%	54%	54%	54%
Random Forest	56%	57%	56%	57%
Neural Networks	53%	54%	53%	54%

Figure 29 – Scenario 2 scores

10. **Scenario 3** – The third scenario was a classification problem like the first one, with the difference that data was filtered excluding features with correlation higher than 0.8 and p-value less than 0.05. In this case one also found **very poor performance** for all models. The models perform better than in comparable first scenario, allowing to conclude that the models behave better excluding correlated features and low p-value.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Gaussian Naive Bayes	41%	38%	49%	36%
K Nearest **Neighbors**	38%	32%	39%	33%
Logistic Regression	44%	49%	59%	40%
Decision Trees	43%	33%	43%	31%
Support Vector Machine	31%	27%	38%	18%
Multinomial Naive Bayes	43%	44%	59%	39%
Random Forest	36%	30%	36%	32%
Neural Networks	36%	32%	36%	33%

Figure 30 – Scenario 3 scores

11. **Scenario 4** – The fourth scenario like third scenario but considering a binary classification. As expected, the performance was **better than in third scenario**, and it allows support the previous conclusion that in this kind of problem it is better to have a binary classification problem than with multiple values in the target variable.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Gaussian Naive Bayes	56%	57%	56%	52%
K Nearest **Neighbors**	53%	52%	53%	50%
Logistic Regression	57%	58%	57%	53%
Decision Trees	57%	58%	57%	52%
Support Vector Machine	47%	47%	100%	64%
Multinomial Naive Bayes	57%	57%	57%	53%
Random Forest	53%	53%	53%	53%
Neural Networks	56%	56%	56%	54%

Figure 31 – Scenario 4 scores

12. **Scenario 5** – The fifth scenario was a classification problem, with pre-processing text data, but with features extraction made using Bag of Words and TF-IDF. At this stage of the analysis, it was decided to exclude the poorest performance models of the first four scenarios and only considered the four best performers in the previous scenarios. Even

that is a classification problem it **performed better than all the other models**, allowing to conclude that, in this kind of problems, the features extraction using Bag of Words and TF-IDF is better than the counting features extraction done for the first four scenarios.

Model	Accuracy	Precision	Recall	F1 Score
Gaussian Naive Bayes				
K Nearest **Neighbors**				
Logistic Regression	60%	64%	66%	57%
Decision Trees				
Support Vector Machine	67%	67%	67%	65%
Multinomial Naive Bayes	43%	60%	58%	35%
Random Forest	68%	66%	69%	63%
Neural Networks				

Figure 32 – Scenario 5 scores

13. **Scenario 6** - The sixth scenario was like the fifth scenario with the difference that considers a binary classification problem, were the non-binary results were excluded, with pre-processing text data, but with features extraction was made using Bag of Words and TF-IDF. As expected, the performance was better than in fifth scenario, therefore **the best performance of all scenarios**.

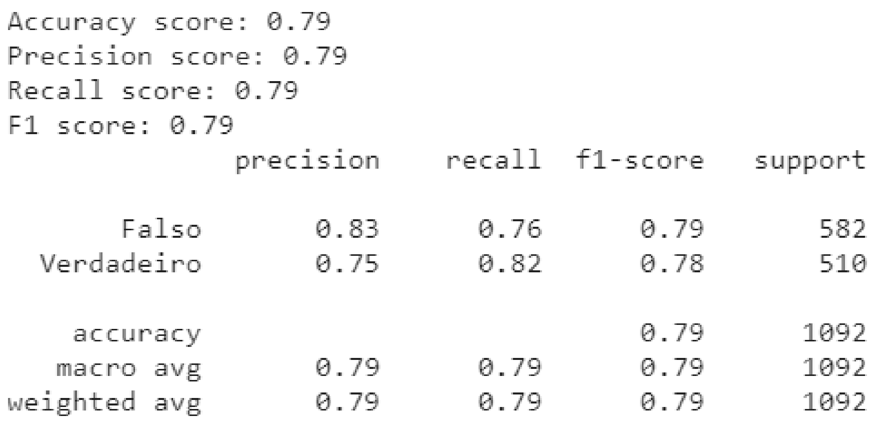
Model	Accuracy	Precision	Recall	F1 Score
Gaussian Naive Bayes				
K Nearest **Neighbors**	65%	66%	65%	64%
Logistic Regression	77%	77%	77%	77%
Decision Trees				
Support Vector Machine	86%	86%	86%	86%
Multinomial Naive Bayes	57%	67%	57%	46%
Random Forest	84%	84%	84%	84%
Neural Networks				

Figure 33 – Scenario 6 scores

After the scenario's execution, we can conclude that sixth scenario is the one that has the best models, considering the scores analyzed. The best three models were Logistic Regression, SVM and Random Forest, for a binary problem with features extraction using Bag of words and TF-IDF, therefore these are de ones that will be deeper analyzed and potentially fine-tuned.

3.7.2. Logistic Regression Results

For logistic regression model we found an average score 0.79 for the four metrics, indicating a good performance.



```
Accuracy score: 0.79
Precision score: 0.79
Recall score: 0.79
F1 score: 0.79
              precision    recall  f1-score   support

       Falso       0.83      0.76      0.79       582
  Verdadeiro       0.75      0.82      0.78       510

    accuracy                           0.79      1092
   macro avg       0.79      0.79      0.79      1092
weighted avg       0.79      0.79      0.79      1092
```

Figure 34 – Scenario 6 - Logistic Regression Scores

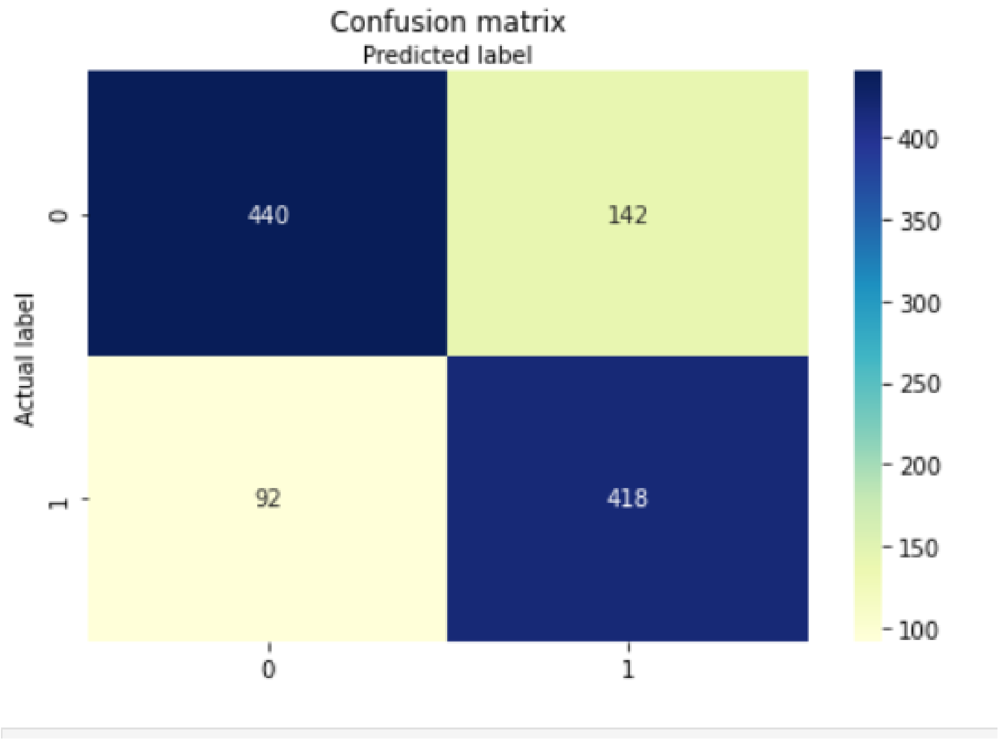


Figure 35 – Scenario 6 - Logistic Regression Confusion Matrix

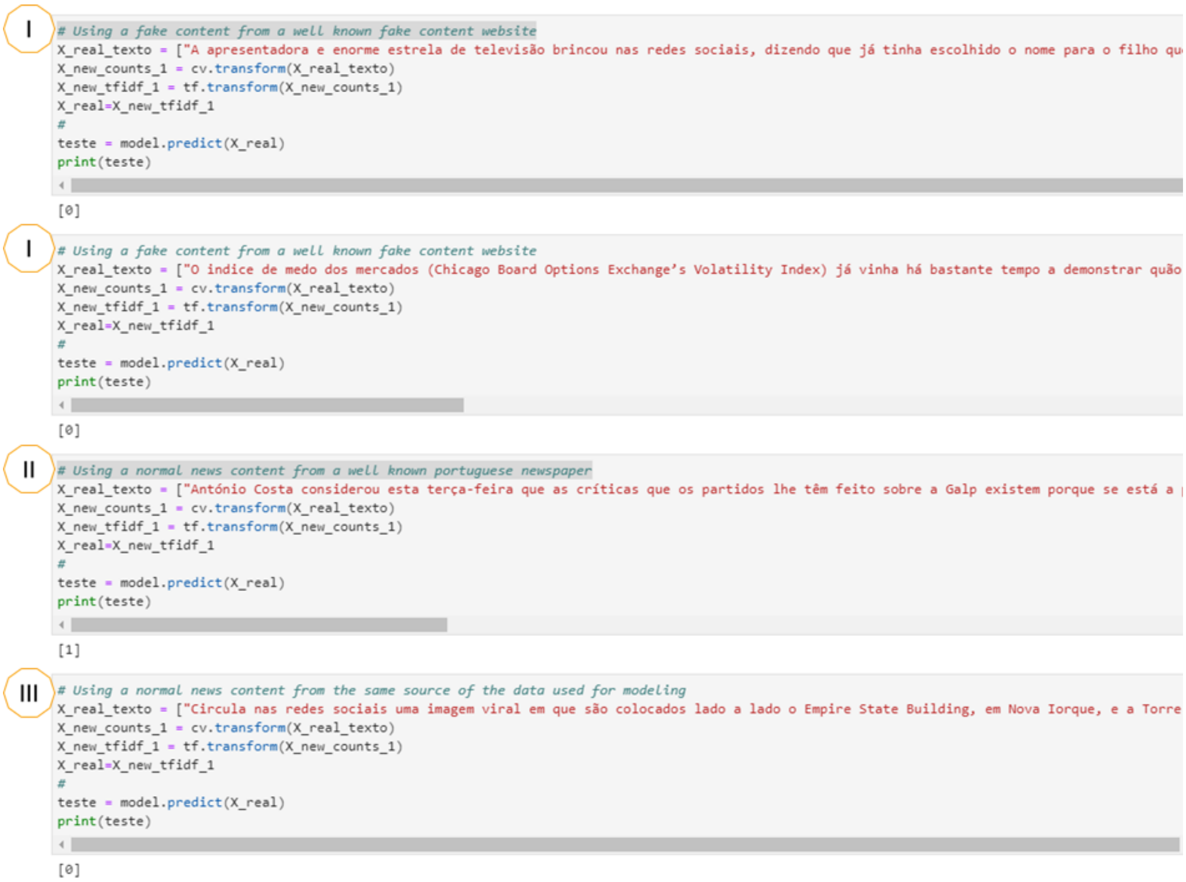
For a better understanding of the model's performance, new data gathered from diferent sources has been presented to the model and the results were:

I. With information gathered from a well known fake news web site (https://www.bombeiros24.com/), the two predictions **returned false correctly**;

II. With a recent article (October 2021) from a newspapper website (https://www.tsf.pt/), it **predicted correctly as True**;

III. With a fact-check article (Ocotber 2021) (https://poligrafo.sapo.pt/fact-check/empire-state-building-e-torre-eiffel-foram-iluminados-em-apoio-as-manifestacoes-convocadas-por-bolsonaro) from the same source of the data used for modeling, it **predicted False correctly**.

Below we can find the evidence of the new text data that were presented to the model and the execution result:



```
# Using a fake content from a well known fake content website
X_real_texto = ["A apresentadora e enorme estrela de televisão brincou nas redes sociais, dizendo que já tinha escolhido o nome para o filho qu
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)

[0]
```

```
# Using a fake content from a well known fake content website
X_real_texto = ["O indice de medo dos mercados (Chicago Board Options Exchange’s Volatility Index) já vinha há bastante tempo a demonstrar quão
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)

[0]
```

```
# Using a normal news content from a well known portuguese newspaper
X_real_texto = ["António Costa considerou esta terça-feira que as críticas que os partidos lhe têm feito sobre a Galp existem porque se está a
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)

[1]
```

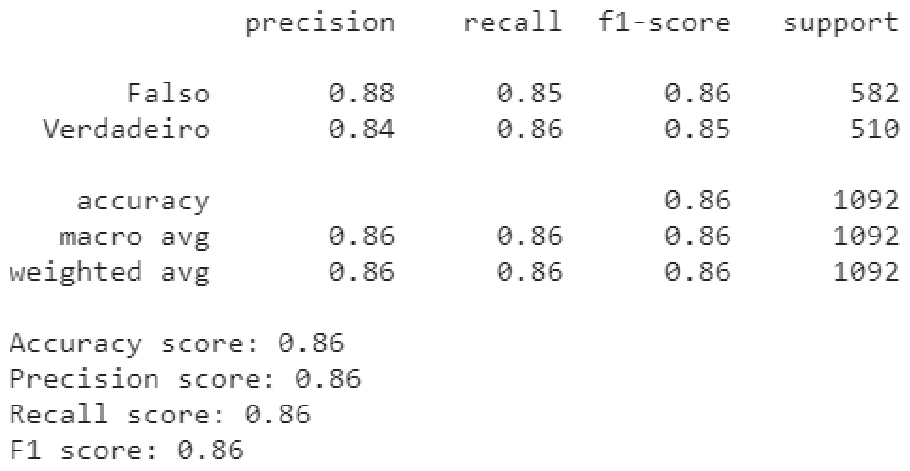
```
# Using a normal news content from the same source of the data used for modeling
X_real_texto = ["Circula nas redes sociais uma imagem viral em que são colocados lado a lado o Empire State Building, em Nova Iorque, e a Torre
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)

[0]
```

Figure 36 - Scenario 6 - Logistic Regression Test Results

3.7.3. Support Vector Machines Results

For support vector machines model, we found a 0.86 average score for the four metrics, indicating a good performance.



```
              precision    recall  f1-score   support

       Falso       0.88      0.85      0.86       582
  Verdadeiro       0.84      0.86      0.85       510

    accuracy                           0.86      1092
   macro avg       0.86      0.86      0.86      1092
weighted avg       0.86      0.86      0.86      1092

Accuracy score: 0.86
Precision score: 0.86
Recall score: 0.86
F1 score: 0.86
```

Figure 37 – Scenario 6 - Support Vector Machines Scores

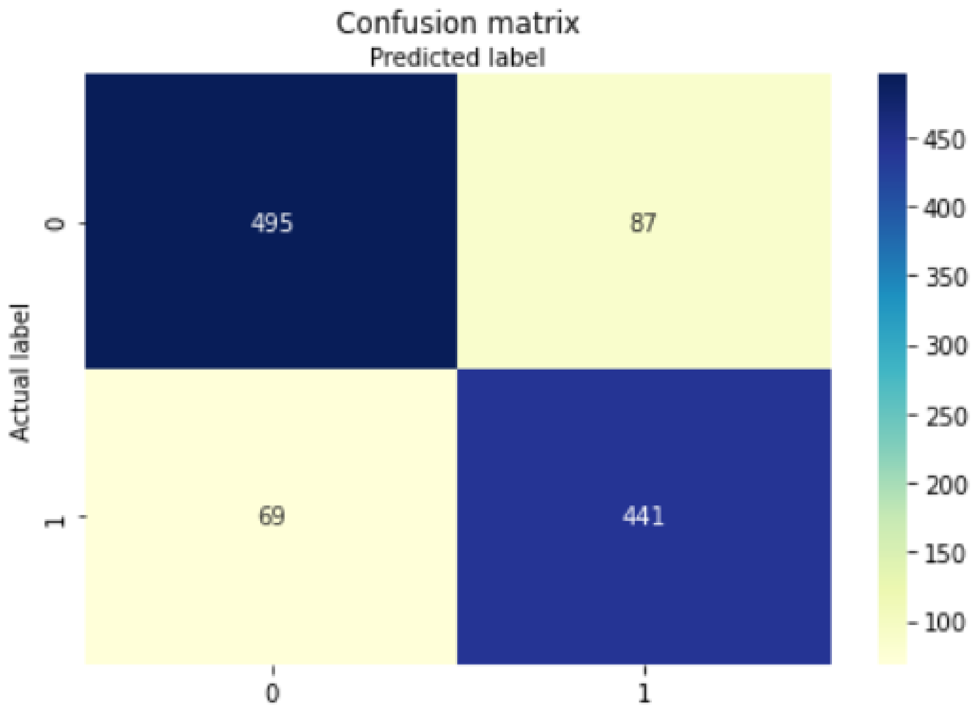


Figure 38 – Scenario 6 - Support Vector Machines Confusion Matrix

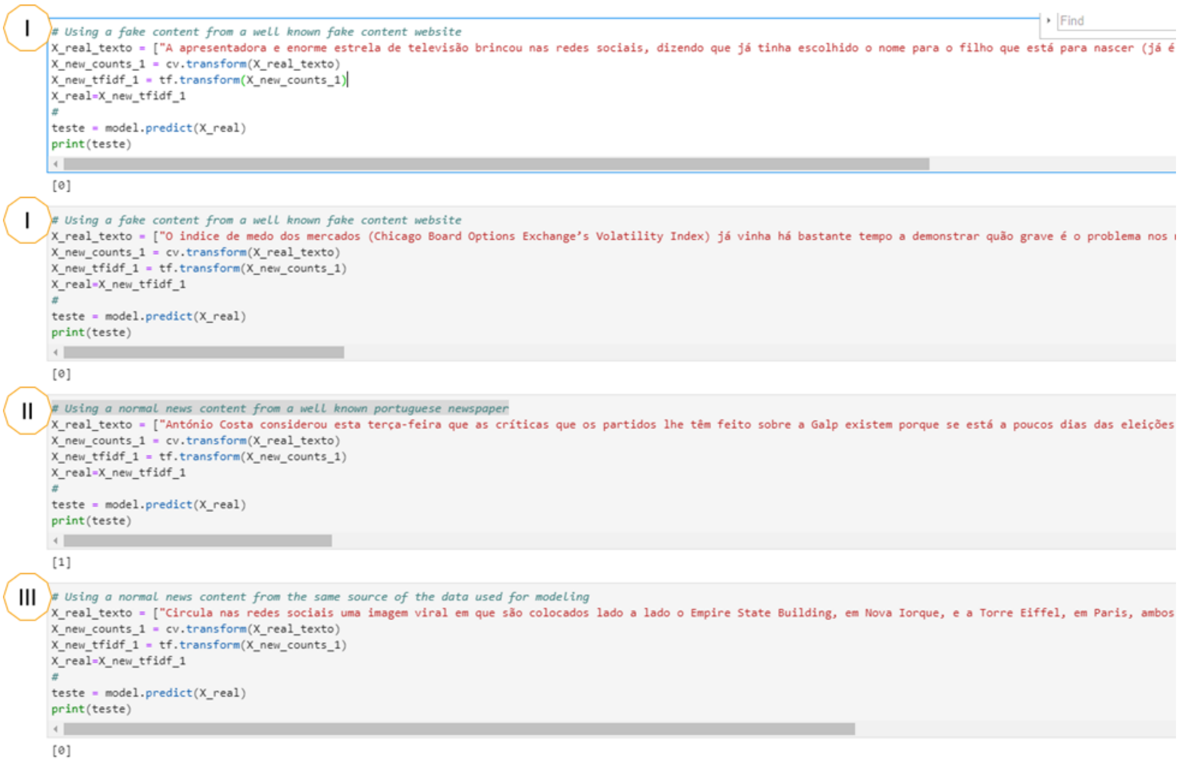
For a better understanding of the model's performance, new data gathered from diferent sources has been presented to the model and the results were:

I. With information gathered from a well known fake news web site (https://www.bombeiros24.com/), the 2 predictions **returned false correctly**;

II. With an article (October 2021) from a newspapper website (https://www.tsf.pt/), it **predicted correctly as True**;

III. With a fact-check (Ocotber 2021) article (https://poligrafo.sapo.pt/fact-check/empire-state-building-e-torre-eiffel-foram-iluminados-em-apoio-as-manifestacoes-convocadas-por-bolsonaro) from the same source of the data used for modeling, it **predicted False correctly**.

Below we can find the evidence of the new text data that were presented to the model and the execution result:



```
# Using a fake content from a well known fake content website
X_real_texto = ["A apresentadora e enorme estrela de televisão brincou nas redes sociais, dizendo que já tinha escolhido o nome para o filho que está para nascer (já é
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[0]
```

```
# Using a fake content from a well known fake content website
X_real_texto = ["O indice de medo dos mercados (Chicago Board Options Exchange's Volatility Index) já vinha há bastante tempo a demonstrar quão grave é o problema nos
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[0]
```

```
# Using a normal news content from a well known portuguese newspaper
X_real_texto = ["António Costa considerou esta terça-feira que as críticas que os partidos lhe têm feito sobre a Galp existem porque se está a poucos dias das eleições
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[1]
```

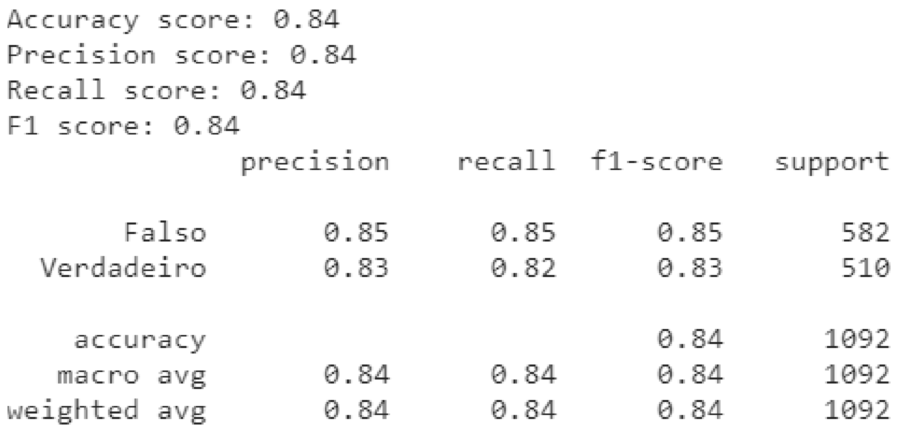
```
# Using a normal news content from the same source of the data used for modeling
X_real_texto = ["Circula nas redes sociais uma imagem viral em que são colocados lado a lado o Empire State Building, em Nova Iorque, e a Torre Eiffel, em Paris, ambos
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[0]
```

Figure 39 - Scenario 6 - Support Vector Machines Test Results

3.7.1. Random Forest Results

For this model we found an average score of 0.84 for the four metrics, indicating a good performance.



```
Accuracy score: 0.84
Precision score: 0.84
Recall score: 0.84
F1 score: 0.84
              precision    recall  f1-score   support

       Falso       0.85      0.85      0.85       582
  Verdadeiro       0.83      0.82      0.83       510

    accuracy                           0.84      1092
   macro avg       0.84      0.84      0.84      1092
weighted avg       0.84      0.84      0.84      1092
```

Figure 40 – Scenario 6 - Random Forest Scores

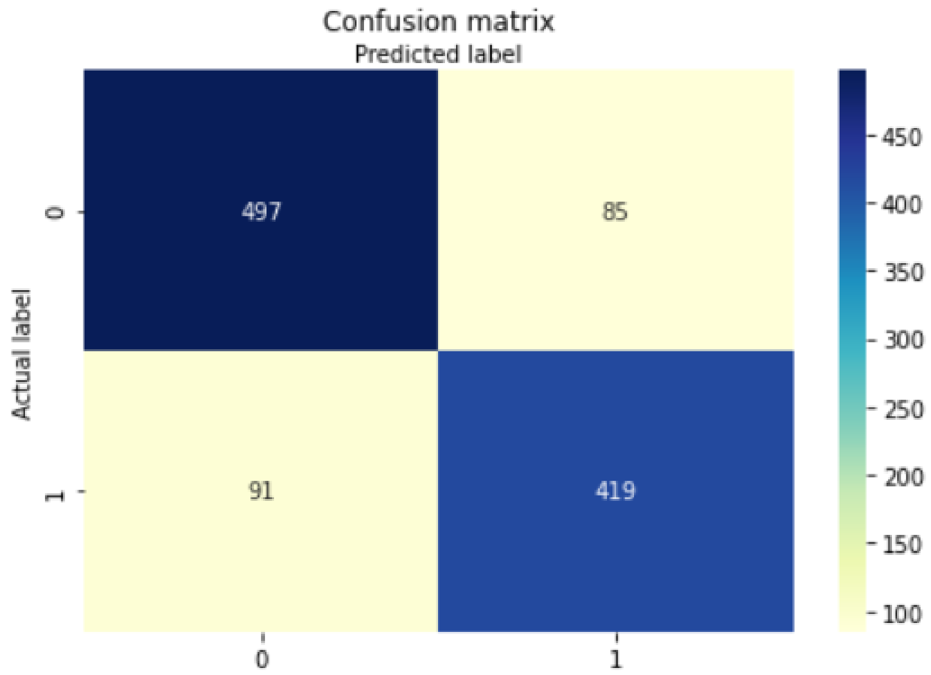


Figure 41 – Scenario 6 - Random Forest Confusion Matrix

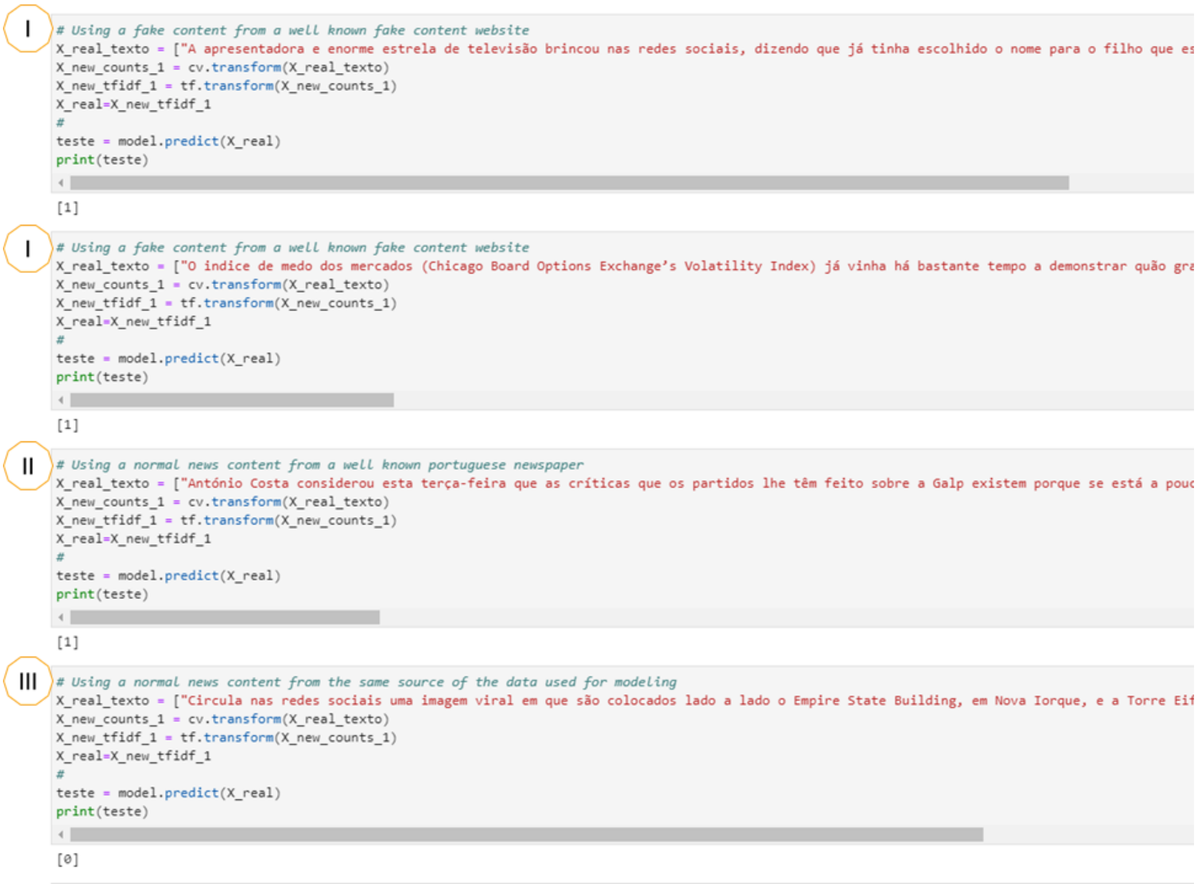
For a better understanding of the model's performance, new data gathered from diferent sources has been presented to the model and the results were:

I. With information gathered from a well known fake news web site (https://www.bombeiros24.com/), the 2 predictions **returned true incorrectly**.

II. With an article (October 2021) from a newspapper website (https://www.tsf.pt/) and also **predicted correctly as True**.

III. Wirh a fact-check (Ocotber 2021) article (https://poligrafo.sapo.pt/fact-check/empire-state-building-e-torre-eiffel-foram-iluminados-em-apoio-as-manifestacoes-convocadas-por-bolsonaro) from the same source of the data used for modeling, it **predicted False correctly**.

This model had the worst performance when presented to new data.



```
# Using a fake content from a well known fake content website
X_real_texto = ["A apresentadora e enorme estrela de televisão brincou nas redes sociais, dizendo que já tinha escolhido o nome para o filho que es
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[1]
```

```
# Using a fake content from a well known fake content website
X_real_texto = ["O indice de medo dos mercados (Chicago Board Options Exchange's Volatility Index) já vinha há bastante tempo a demonstrar quão gra
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[1]
```

```
# Using a normal news content from a well known portuguese newspaper
X_real_texto = ["António Costa considerou esta terça-feira que as críticas que os partidos lhe têm feito sobre a Galp existem porque se está a pouc
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[1]
```

```
# Using a normal news content from the same source of the data used for modeling
X_real_texto = ["Circula nas redes sociais uma imagem viral em que são colocados lado a lado o Empire State Building, em Nova Iorque, e a Torre Eif
X_new_counts_1 = cv.transform(X_real_texto)
X_new_tfidf_1 = tf.transform(X_new_counts_1)
X_real=X_new_tfidf_1
#
teste = model.predict(X_real)
print(teste)
```

```
[0]
```

Figure 42 - Scenario 6 - Random Forest Test Results

3.8. Model Fine Tuning

Considering the results of the three models, we can say that, in this case, the best models for this kind of problem can be based on Support Vector Machines algorithm. Therefore, I tried to improve it making some parameter variation using the python functions and parameters available but didn't manage to improve the model performance measures significantly. Therefore, the best model is SVM using a "linear" kernel function. In machine learning, a "kernel" is usually used to refer to the kernel trick, a method of using a linear classifier to solve a non-linear problem. It entails transforming linearly inseparable data like to linearly separable ones. The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

4. CONCLUSION

One can say that the text mining and supervised machine learning modeling can help to spot fake content over the internet. Probably the models must be updated with new content in a regular basis to keep a good performance.

Answering the questions that were the goal of the current work:

- **How do different approaches in the methods and features extraction can affect the predictive performance of models?**

  Like we have seen above, the variations in the process can end in very different results and we found that, for this problem, with this data, the best approach was using a binary classification problem with feature extraction using the SVM modeling.

- **How do we provide additional help so that people can spot fake content in news websites or social media?**

  With additional work on this kind of modeling, with more sophisticated technology and capabilities, with regular updates, there could be developed an API or other kind of interface, for example integrated with Facebook, that would call the predict function and receive an answer if a given text is true or false. This approach should be made carefully because people might not be aware that this is a prediction, that in some cases will fail and it's not a deterministic result.

- **Can fact checking activity be performed automatically by machines?**

  As we have seen in the models that were experimented, they are not 100% accurate and can fail the prediction, meaning that we will not be able to turn this task totally automated, but if the fact checking teams have one thousand articles to validate, they can run the prediction and start by the ones that are fake, considering that those are the ones that promote bad things in our society like misinformation and manipulation. These obviously considering other parameters like the number of shares or retweets, so that they work on the most impacting ones.

Further investigation can be made using more complex feature extraction tools and using also more complex models. One of the gaps in this problem was that many of the texts feature extraction and modeling functions are only available for English language, and that was the reason why sentiment analysis features weren't used for modeling, because the text was translated using Google Translator, losing some specific semantics from the Portuguese language and then calculated the sentiment scores.

Additionally, this work might be a helpful tool for fact-checkers in Portugal and for example can be shared with the fact checkers from Poligrafo for them to upgrade and use if they want. Also, other researchers from the academic institutions can use the works insights to evolve for better and more performant models.

5. BIBLIOGRAPHY

Banerjee, Snehasish, Alton Y. K. Chua, and Jung Jae Kim. 2015. “Using Supervised Learning to Classify Authentic and Fake Online Reviews.” *ACM IMCOM 2015 - Proceedings*.

Bentley, Jon Louis. 1975. “Multidimensional Binary Search Trees Used for Associative Searching.” *Communications of the ACM* 18(9).

Billy Perrigo. 2020. “Facebook’s Hate Speech Algorithms Leave Out Some Languages | Time.” Retrieved June 24, 2020 (https://time.com/5739688/facebook-hate-speech-languages/).

Brian Fung. 2020. “Twitter Labeled Trump Tweets with Fact-Check Labels for the First Time - CNN.” Retrieved June 24, 2020 (https://edition.cnn.com/2020/05/26/tech/twitter-trump-fact-check/index.html).

Cardoso, Emerson F., Renato M. Silva, and Tiago A. Almeida. 2018. “Towards Automatic Filtering of Fake Reviews.” *Neurocomputing* 309:106–16.

Chang, Jon M. 2010. “Samsung.” *ABCNews* 68:855–60.

Euronews. 2021. “No Title.” Retrieved (https://www.euronews.com/2021/10/21/donald-trump-to-launch-social-network-after-being-banned-by-twitter-facebook-and-youtube).

Hankey, Stephanie. 2020. “The Behavioral Data Debate We Need by Stephanie Hankey - Project Syndicate.” Retrieved June 29, 2020 (https://www.project-syndicate.org/commentary/covid19-digital-technologies-behavioral-data-debate-by-stephanie-hankey-2020-06).

Hilder, Paul Lewis and Paul, and hilder paul. 2018. “Leaked: Cambridge Analytica’s Blueprint for Trump Victory | UK News | The Guardian.” *The Guardian* 1–5.

Klein, Ewan. 2006. “Computational Semantics in the Natural Language Toolkit.” *Technology*.

Li, Huayi, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. 2014. “Spotting Fake Reviews via Collective Positive-Unlabeled Learning.” *Proceedings - IEEE International Conference on Data Mining, ICDM* 2015-Janua(January):899–904.

Li, Nan, and Desheng Dash Wu. 2010. “Using Text Mining and Sentiment Analysis for Online Forums Hotspot Detection and Forecast.” *Decision Support Systems* 48(2):354–68.

Luz Yolanda Toro Suarez. 2015. “Study: 70% of Facebook Users Only Read the Headline of Science Stories before Commenting.” 1–27.

Manzoor, Syed Ishfaq, Jimmy Singla, and Nikita. 2019. “Fake News Detection Using Machine Learning Approaches: A Systematic Review.” *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019* (Icoei):230–34.

Omohundro, Stephen M. 1989. “Five Balltree Construction Algorithms.” *Science* 51(1).

Parikh, Shivam B., and Pradeep K. Atrey. 2018. “Media-Rich Fake News Detection: A Survey.” *Proceedings - IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018* 436–41.

Pedregosa, Fabian, Ron Weiss, Matthieu Brucher, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12:2825–30.

Polígrafo. n.d. “Poligrafo.Pt.” Retrieved (poligrafo.pt).

Reyes-Menendez, A., J. R. Saura, and F. Filipe. 2019. “The Importance of Behavioral Data to Identify Online Fake Reviews for Tourism Businesses: A Systematic Review.” *PeerJ Computer Science* 2019(9).

Richardson, Leonard. 2016. “Beautiful Soup Documentation.” *Media.Readthedocs.Org*.

Shahi, Gautam Kishore, Anne Dirkson, and Tim A. Majchrzak. 2020. “An Exploratory Study of COVID-19 Misinformation on Twitter.” *ArXiv*.

Shane, Tommy. 2020. “The Psychology of Misinformation: Why We’re Vulnerable.” *First Draft*.

Tardáguila, Cristina. 2020. “International Fact-Checkers Aren’t Quite Celebrating Zuckerberg’s Decision to Block Trump.” *Poynter* 1–18.

The Poynter Institute. 2021. “Verified Signatories of the IFCN Code of Principles.” *IFCN Code of Principles* 1–13.

Tylor, Cari-Ann. 2015. “Chef Sacked after Putting Negative Revi...out Rivals on TripAdvisor _ Metro News.Pdf.”

Wikipedia. 2022 “Logistic Function.” Retrieved (https://en.wikipedia.org/wiki/Logistic_function).

Wikipedia. 2022 “Support-Vector Machine.” Retrieved (https://en.wikipedia.org/wiki/Support-vector_machine).