People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Abderrahmane Mira University of Bejaia

Exact Sciences Faculty
Computer Science Departement

Submitted in partial fulfilment of the requirements

for the degree of
master of computer science option artificial intelligence

Detecting and monitoring hate speech in tweets

Author
Maria MENNI
Tiziri YOUNSI

The president of the jury
Dr. Soraya ALOUI
Examiner
Pr. Kamal AMROUN
Supervisor
Dr. Houda EL BOUHISSI

promotion 2021_2022

Contents

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface

AUC Area Under Curve

BOW Bag Of Words

CNN Convolutional Neural Network

DL Deep Learning

LR Logistic Regression

LSTM Long Short-Term Memory

ML Machine Learning

NB Naive Bayes

NLP Natural Language Processing

NN Neural Network

NUMPY NUMerical PYthon

OM Opinion Mining

ROC Receiver Operating Characteristic

SA Sentiment Analysis

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document Frequency

WE Word Embedding

Acknowledgement

Praise and thanks to God, the Almighty, for His blessings throughout our research work.

We would like to express our gratitude to our supervisor Mrs.ELBOUHISSI, who has always been attentive and available throughout the realization of this thesis. We thank her for having framed, guided and advised us, as well for the inspiration, the help and the time that she has devoted for us.

We would also like to thank all the professors, speakers and all those who, in one way or another, have contributed to the success of this work and who could not be cited here.

Finally, we would like to address our sincere gratitude for each of the members of the jury for their interest in this work and agreeing to evaluate it.

Dedication

We dedicate this work to our families: our parents, our grandparents, our sisters and brothers, who accompanied, helped, supported and encouraged us throughout the realization of this dissertation.

We also dedicate this dissertation to our friends, who shared with us this journey.

Chapter 1

General Introduction

During the last three years of the pandemic, the lock downs had a serious impact on people's behavior. They become more oppressed and willing to express their feelings through social media. However, this virtual communication has also been increasingly exploited for the propagation of hate speech in tweets[2]. The discrimination of certain categories of society has always existed through different forms, however the growth of social media use has encouraged the spread of this phenomenon. Hate speech can be defined as any use of aggressive, violent or offensive words or expressions against a person, or group of people based on specific characteristics such as race, color, ethnicity, gender, nationality, religion or other characteristics [31]. The propagation of offensive tweets is leading to serious consequences, not only on individuals affecting their mental health but also on societies causing conflicts between communities; studies show that most crimes committed in the real world are directly related to online hate speech.

Therefore, many countries and organizations are establishing laws to prohibit hate speech in tweets. Social media services such as Facebook, Instagram and twitter are aiming to improve their methods of detecting and treating these tweets by applying regulation policies, built up on their own definitions to limit these offensive comments without violating the right to freedom of expression.

However, the identification of text containing hate words in tweets is still a challenging task for both humans and machines. In many cases, detecting texts referring to hateful content without containing offensive words is a huge problem to solve, since the meaning of the content can refer to toxic messages without being classified as hate speech tweets[10]. Therefore, an automated method for detecting hate speech in tweets is needed. Multiple works, using natural language processing techniques in combination with machine learning (ML) methods and deep learning (DL), have been devoted to detecting whether a tweet in a social network is considered as hateful

or not. The detection of hate speech in tweets can serve several purposes, including: the fight against cyberbullying, as well as crime, racist and misogynistic insults and the identification of regions with high crime rates.
Recently, developers are becoming increasingly interested in sentiment analysis of texts due to the spread of online communication through the Internet, such as social media, email, and forums. Besides, recently, natural language processing (NLP) and text mining techniques have improved considerably with the recent advances in machine learning. Creative use of advanced artificial intelligence techniques, in particular, the use of machine learning has become progressively popular to support content moderation in online platforms.
Our work is a step in that direction; it is devoted to exploring machine learning and sentiment analysis methods for the automated detection of harmful content in social media.

1.1 Problem Statement

One of the biggest impacts of the widespread use of social networks is the increase of hate speech in comments, spreading hateful and aggressive messages to individuals and/or groups of society can not only cause harm to individuals leading in some cases to self harm , mental health troubles and in some cases suicide but also harm on a bigger scale and conflicts between communities.
The interpretation of human language by machines is a complex task that social platforms are aiming to solve by improving developed methods and models to limit the aggressive content without violating the right to freedom of expression. Our key challenge is to improve the efficiency of already existing approaches for the purpose of identifying and classifying the tweets into hate speech and non-hate speech by applying artificial intelligence methods on twitter data sets .

1.2 Goal

The main objective of this work is to build an approach in order to identify offensive text that may be contained in posts published in Twitter based on machine learning algorithms. Therefore, multiple pre-processing techniques, feature generation and classification algorithms are combined and applied on social media dataset. Our goal is to determine which method gives the best results for the detection of hate speech in tweets.

1.3 Thesis organization

The rest of this thesis is structured as follow: The second chapter is devoted to the definition of the domain, we will be presenting the different concepts related to the detection of hate speech, in addition to the several methods and techniques of artificial intelligence.

In the third chapter, we elaborated a statement of art where multiple works were discussed and categorized into two major classes based on their approach. We established a comparative table for the studied works resuming each article. Finally we analyzed the different approaches seen and compared it to our proposed approach.

The fourth chapter deals with the experimentation of our approach. It presents the different aspects related to the implementation of the prototype that we will develop and the different phases of our conception.

The fifth chapter presents the programming tools, the implementation of interfaces and the results of experiments, as well as the software tool chosen for the implementation of our approach.

Finally, we conclude this thesis with a general conclusion, assessing the elaborated work and we propose a set of perspectives for future works.

Chapter 2

Generalities and basic concepts

2.1 Introduction

Toxic content has shown a tendency to increase in recent years on social media and is becoming one of the major problems in the world. Manual techniques to detect hate speech in comments are no longer effective, thus, the need to develop tools to detect offensive content is essential.

In order to understand the context of our work and the goal of our study, we will be introducing the different concepts related to our project based on multiple research articles.

This chapter is divided into two main sections. The first one is a presentation of the concept of hate speech and its types. In the second chapter, we explain the notions of feature extraction and the different methods of text classification (machine learning, deep learning methods and Natural Language Processing).

2.2 Hate speech in social media

2.2.1 Definition of hate speech

There is no standard definition for hate speech, authors have proposed different interpretations of the hate speech concept.
[5],[1] defined hate speech as any sort of communication or exchange of words, expressions, images or videos in a particular language inciting hate, violence and aggressive or any form of discrimination towards individuals or group of persons based on their gender, religion, ethnicity, race or other characteristics .
Hate speech can be classified in different categories:

2.2.1.1 Gendered Hate speech

The assumption that a person is more important than another based on its gender. It often leads to discrimination against the members of the assumed inferior gender. This category is exposed to harassment and devaluation in social media posts [1].

2.2.1.2 Religious hate speech

The discrimination of groups of society based on their religions and beliefs, engendering a climate of violence or intolerance between communities. This may lead in some cases to serious consequences (suicidal attacks).

2.2.1.3 Racist hate speech

Any sort of differentiation in the treatment of individuals or communities on the bases of their race, ethnicity or color. For example, disrespecting an individual because of his membership to a specific race or region.
The following table will illustrate the different types of hate speech:

Type	Example
Gender	“Women are dumb.”
Religion	“Islam out of Britain. Protect the British people".
Color	“Blacks are inherently inferior, lecherous, predisposed to criminal activities, and should not be allowed to move into respectable areas.”
Ethnicity	“Arabs out of France.”
political	“ We have a stupid government, devide and rule is their motto.”

Table 2.1: Hate Speech types and examples.

2.3 Concepts

2.3.1 Features extraction methods

An important step, where raw data is transformed into numerical features more manageable to process by the machine learning algorithms.
In this section, we will explain some of the most common approaches to extract

features.

2.3.1.1 Bag of words (BOW)

The most used text representation method, it consists in representing the document text as a vector where each dimension represents a particular word, and the value could represent either the frequency of the word in the document or its occurrence (1 or 0), or other values. The notation of BOW goes to the representation of the document as a bag of its words[13].

2.3.1.2 Term Frequency-Inverse Document Frequency (TF-IDF)

A measurement method to estimate the importance of a word in a document among a collection of documents.
TF: is evaluated by counting the frequency of a particular word in a document, this frequency can be adjusted by the length of the document.
IDF: it allows determining how much a word is common or rare in the entire corpus, by dividing the number of the documents by the number of documents where the word appears.
TF-IDF is calculated by multiplying the two values (TF and IDF)[13].

2.3.1.3 Word Embedding

A text representation method that allows representing words that have the same semantic by similar representations (numeric vectors), for example, the words "dog" and "cat" could be represented by vectors that are close in the vector space since the two words are semantically the same.
This method has multiple models, the most common are:

- **Word2Vec :** developed by Google, it produces a vector space containing words that have a common context. This approach has two main methods (Continuous BOW: starting from a word to predict the context and Skip-Gram: starting from the context to predict the word).
- **GloVe :** an extension of Word2Vec developed by Stanford, using the word co-occurrence matrix to generate word embedding.
- **FastText:** a word embedding method developed by Facebook, another extension of Word2Vec, where each word is represented as an n-gram of characters.

Word embedding combined with machine learning methods or deep learning approaches are greatly used for the detection of hate speech, and many other challenges such as sentiment analysis.

2.3.2 Text classification techniques

2.3.2.1 Machine learning algorithms (ML)

A field of study of artificial intelligence that permits computers to learn and think on their own without being programmed. ML is used for many tasks such as prediction and pattern recognition[5].
Machine learning can be classified into two categories:

- Supervised learning:
  A set of training data is given as input with the desired result during the learning phase, the algorithm applies the learned model on the outputs and compares them with the correct results. Supervised learning can be classified as classification if the outputs are discrete and regression if they are continuous [3].

- Unsupervised learning:
  The training data contains only unlabeled data with no correct results, the concept of this approach is to find a model based on the recognition of the patterns in the input data (clustering). This method is efficient in the classification of unlabeled data[3].

In our case, as the detection of hate tweets is a classification problem, we will focus on supervised learning algorithms.

Support Vector Machine (SVM)

Support Vector Machine is a supervised ML algorithm mainly dedicated to solving classification tasks. The concept of this algorithm is to find a hyperplane in N- Dimensional space (N is the number of features) to separate data points into different classes[27].
The main objective is to find, between all existing plans, a hyperplane that has a maximum margin; which means that the distance between the support vectors (the data points of both classes closer to the hyperplane) must be maximum to build a robust SVM algorithm and easily classify the new data points.

Naive Bayes (NB)

Naive Bayes classifier is popular to be one of the fast and powerful ML algorithms for classification classes of datasets, besides, it has better results compared to other algorithms in multi-class classifications. It is a probabilistic classifier that utilizes Bayes theorem [32]:

$$P(A|B) = P(B|A)P(A)/P(B). \tag{2.1}$$

We use this algorithm to assign a hypothesis h (can be a class) to a new data d in classification problems. We suppose that the attributes are conditionally independent and unrelated relatively to each other. From which we deducted the concept of naivety in 'Naive Bayes'[23].

Logistic Regression (LR)

Logistic Regression is considered one of the most important techniques in ML discipline for its power in simplifying complex statistical calculations. It is an algorithm dedicated mainly to binary classification problems contrary to linear regression that is used to solve regression problems. It predicts binary output of a categorical dependent variable based on one or several independent variables (x) to classify the data coming into two classes that means the result of the prediction is a probabilistic value (between 0 and 1) which helps to make a decision easily between two alternatives.
Logistic Regression uses a logistic function which is:

$$S(x) = \frac{1}{1 + e^{-x}}$$

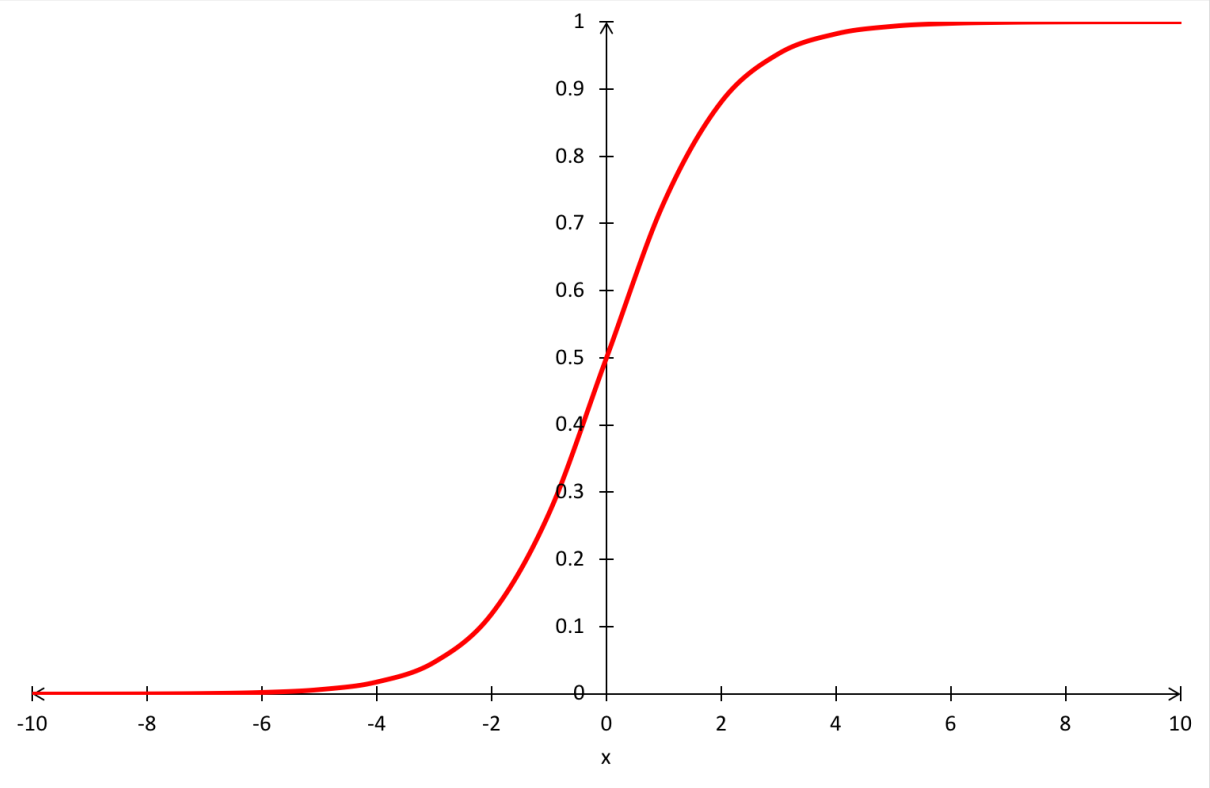

Figure 2.1: Sigmoid function used by LR algorithm.

to build a robust model to predict the outcomes of variables[16].

2.3.2.2 Deep Learning (DL)

A type of machine learning that allows machines to learn how to imitate human behavior. Deep learning methods have improved the process of collecting and analyzing great amounts of data. In The last decade, DL has known a special attention to solve the problem of text classification, in our case the detection of hate speech through many methods and approaches (CNN, LSTM...) that will be addressed later.

NEURAL NETWORK (NN)

Neural Network or Artificial Neural Network is a sub-field of ML discipline and the heart of DL algorithms. The conception of NN was inspired by the functioning of the biological neural network, thus, to imitate human brain behavior, several neurons (or nodes) were interconnected to each other to program computers so they

can make decisions, solve classification or recognition problems. NN performs better when the data (inputs) are massive and varied because the algorithm learns with experience.
ANN has three important components : input layer which accepts all types of data , hidden layer where the entries are examined and all calculations done, finally, the output layer which returns the results of the prediction using the output of the hidden layer[24].

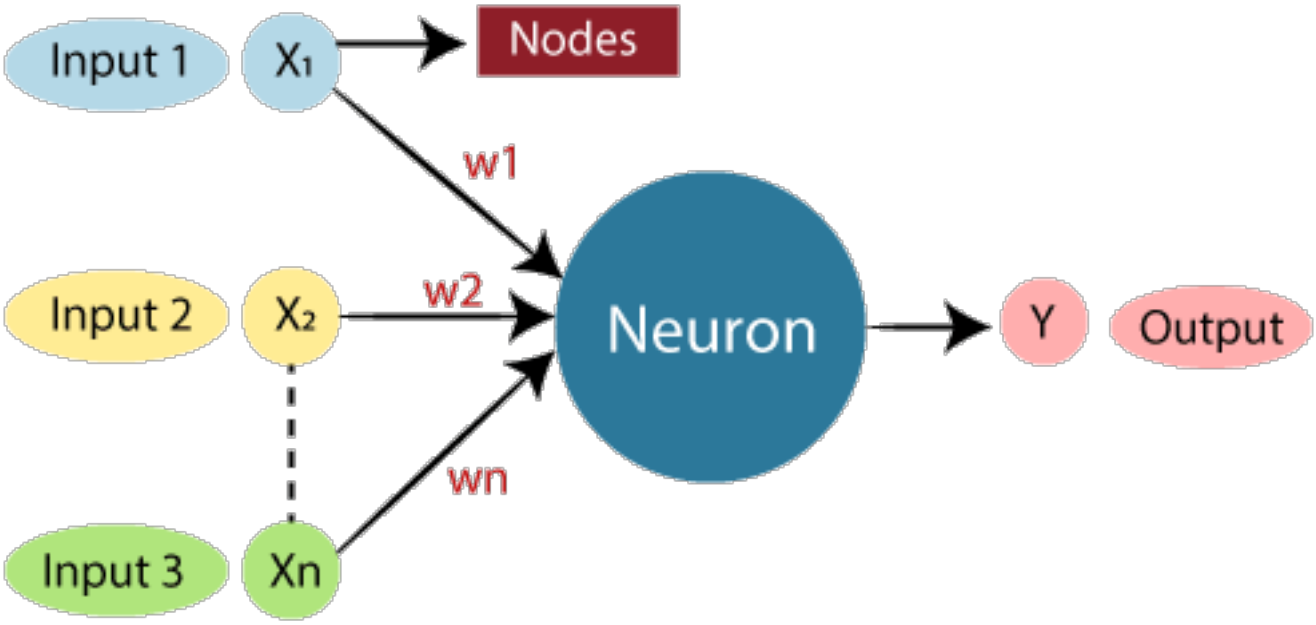

Figure 2.2: Architecture of artificial neural network

Convolutional Neural Network (CNN)

Is a deep learning method to extract features that differs from the traditional manual extraction, it has become one of the most used methods to solve machine learning problems most commonly image recognition but also text classification.
CNN architecture is composed of three principle types of layers:

- Convolution layers:
  An important part of the CNN architecture that is in charge of the extraction of features, by applying a kernel (an array of numbers of size 3x3 or 5x5) on the input (array of numbers called tensor) to obtain a feature map as a result of the operation. Each different kernel is considered as a feature extractor since; using different kernels produces other feature maps[26].

- Pooling layers:
  The main objective is to reduce the dimensionality of the feature thus, the computational complexity of the model. The pooling operation consists in applying a sort of filter to the feature maps; the filter size needs to be smaller than the feature maps (generally 2x2) with a stride of two, meaning each feature map will be reduced by a half.
  One of the common functions of pooling operation is the MAX pooling, where for each patch of the feature map the maximum number is extracted[9].

- Fully connected layers:
  The input of this final layer is the output of the final convolution or pooling layer that is flattened (the vectorization of the output matrix) then linked to the fully connected layer.

Long Short-Term Memory (LSTM)

A recurrent neural network method that solves the problem of short-term memory without losing information through the different steps of the process of RNN. LSTM is used for many different tasks (machine translation, text recognition...).
LSTMs architecture is represented as a sequence chain of cells transporting information, each cell is composed of a set of gates that manages the propagation of information; they are in charge of controlling which information to keep and to dispose, depending on its importance during the training.

2.3.2.3 Natural Language Processing (NLP)

Natural Language processing is a puissant tool of Artificial Intelligence (AI) and a subfield of computer science and linguistics. It is a component that allows computers to understand human language whether it is written or spoken.
NLP is used in several domains of AI for example text classification, text extraction and machine translation.
Its main objective is to facilitate the interaction between humans and computers by taking inputs in reel world and converting them into a specific code so the machines can manipulate and understand the natural language[21].

2.3.2.4 Sentiment analysis in text classification

We consider Sentiment Analysis (SA) or in another term Opinion Mining (OM) as a subfield of NLP. It designates the extraction of sentiments in a user's comment in social media to determine whether it was positive or negative text by analyzing

the emotions behind it.
Given the growth in the use of social media in recent years, it became almost impossible to detect toxic comments manually, thus, this strategy was implemented to successfully assign tags (or labels) to classify them into categories .
To extract the sentiments of a random comment in social media we define its sentiment polarity which is a float between +1 and 1 (-1 : a positive statement and -1 a negative one), this metric allows us to define the nature of the judgments and emotions expressed by the users.

2.4 Conclusion

In this second chapter, we have introduced the idea of hate speech and its different types illustrated by examples. Then, we presented feature extraction techniques. Finally, we approached the different methods of text classification and illustrated their architecture with figures .
In the following chapter, we will present and analyze the existing works related to hate speech detection based on the used approaches.

Chapter 3

State of Art

3.1 Introduction

With the increase of social media, users have become more targeted to face abusive comments, thus, social platforms such as Facebook, Instagram and Twitter are seeking to improve their automated methods of detecting these comments. Due to the complexity of human language, different methods and techniques were explored to improve the performance of machines in interpreting the texts.

In literature, there are several works related to the detection of hate speech in tweets where they have used and compared different methods and techniques of artificial intelligence to solve this problem. To establish the state of art, researchers adopt several techniques to classify previous works by focusing on categorization such as dataset, pre-processing or classification methods used in the approaches.

In this section, we will present the principal related works in the domain of hate speech detection in tweets, and elaborate a comparative table to analyze the different discussed works.

3.2 Related works

To be able to classify the tweets into categories(hate speech or non-hate speech), we need to apply multiple techniques and approaches. Authors have established many works exploring different methods, based on their performance and results, whether machine learning approaches, or deep learning methods. In order to better understand the functioning of the automated detection of hate speech, we have studied multiple works treating this problem.
We have established a classification of the works based on two categories: deep learning based approaches and machine learning based approaches.

3.3 Deep Learning based approaches

In the last few years, deep learning approaches have gained a great attention for the detection of hate speech in tweets. Numerous established works have shifted to use deep learning methods (CNN, LSTM model, RNN...) due to its capacity of learning various features.

Kovas et al.[14]focused in their work on solving the problem of limitation of available data for the detection of hate speech using deep natural language processing combined with convolutional and recurrent layers by leveraging three external resources in order to compose their labed self-contained corpora : the first data is Hasoc 2019 corpus created based on social media tweets in three languages(English, German and Hindi), the second data OLID similar to Hasoc corpus and finally, the Hatebase dataset. The authors proceed to do a text preprocessing on the datasets, before applying the cross validation method in order to train the machine learning methods.[14] choose to work with deep learning methods combining Convolutional and Long Short Term Memory layers as a support for hyper-parameter optimization. In addition to that, they worked with a RoBERTa transformer model as a feature extractor from the text data using the cross validation method, classical machine learning methods were also applied in their work as KNN, Adaptive Boosting,Linear discremination, Simplest Logistic regression, Random Forest and the two class support vector machine. Lastly, they used the FastText classification models. The experiment's results showed that leveraging additional labeled datasets enhanced the performance of the model confirming their hypotheses.

Vijayaraghavan et al.[30] have discussed the importance of social and cultural context features unlike most works in the literature which ignored this dimension of hate speech comments. Therefore, they proposed a deep learning multi-modal model to detect hate speech based on the socio-cultural background of the users. They extracted and applied a fusion of the semantic, cultural and social context features to define the inputs of the deep learning model.After experimenting with several models, they found that deep learning models outperform the traditional classifiers (Support Vector Machine and Linear Regression), also adding social and cultural context to the deep learning modal makes a better results comparing to purely text-based model.

Elouali et al.[6]have created a model for the classification of tweets into hate

speech and non-hate speech written in seven different languages using convolutional neural networks and character level representation. Due to the unavailability of multilingual hate speech datasets, the authors decided to combine existing datasets in different languages following multiple steps resulting in 46173 tweets. They proceed to do a data cleaning where the insignificant information was removed as HTML parts, mentions, URL links, hashtags and special characters and diacritics before applying the model inspired by a CNN architecture for text classification with character level representation. The outcomes illustrated the good performance of the proposed model for the classification of tweets.

Zhang et al.[34]proposed a new method for the detection of hate speech in tweets using deep neural networks combined with convolutional and long short-term memory networks in order to enhance the efficiency of the performance. They created a different dataset by assembling tweets targeting refugees and Muslims where they employed multiple approaches as the mainstream bootstrap approach and the Twitter streaming API on seven public datasets (WZ-L, WZ-S.amt, WZ-S.exp, WZ-S.gb, WZ-LS, DT and RM), followed by a pre-processing phase. The author's model has a particularity from other similar architectures that consists in using the drop-out and max pooling in the purpose of homogenizing learning and pulling out features from LSTM layer. [34] applied a linear SVM model on different sorts of features called basic features (surface features, linguistic features and sentiment features) in addition to enhanced features (TODO). Aiming to solve the overfitting problems due to the use of multiple features, [20] proposed a feature selection by creating four baseline models (SVM, SVM fs, SVM+, SVM fs+) applied on different datasets. This approach confirmed the efficacy of the features selection in improving the performance compared to other works.

3.4 Machine Learning based approaches

As detection of hate comments is classified as a supervised classification task, machine learning algorithms have found great success in the domain and were the most used by researchers to solve this problem.

Manaa and Abdallah[19]have proposed a system to detect hate speech in tweets with machine learning algorithms such as Naive Bayes , Support Vector Machine and Neural Network. They have chosen two datasets to train and test

the classifiers; the first dataset was employed in[15] . It contains 1,600,000 tweets that have been annotated as 50% negative and 50% positive tweets. The second dataset was employed in [33]. It contains 93% of positive tweets and 7% of negative ones. They[19] have used Doc2vec model to extract features and maintain the order information to prepare the inputs of the classifiers.After experimenting with the different classifiers, they found that the best method to detect hate speech in tweets with doc2vec is by neural network algorithm with an accuracy of 92% with the first dataset and 90% with the second.

MacAvaney et al. [18]adopted a multi-view SVM model to classify hate speech in social media.The approach consists to fit every individual linear support vector machine with a type of feature , in the end, the combinaison if those classifiers produces a meta-classifier. In the first place, they [18] have used TF-IDF and N-gram models to extract features from several datasets to collect as much data as possible to train their classifier. The result of the experiments shows that using TF-IDF weights for characters N-grams works better on Facebook dataset. Besides, they[18] found that the approach outperforms other top-ranked works in the literature by 3,96% of accuracy and 2,41% in terms of macro F1, which means that combining multiple SVMs contributes efficiently in detecting and reducing hate speech in social media.

Gaydhani et al.[7]have proposed a solution using machine learning algorithms and N-gram features with TF-IDF weights to classify tweets into three categories : hateful, offensive and clean. Their model was deployed to interact with Twitter applications to collect data comments with Twitter REST API. To build the model,they considered three ML algorithms such as Logistic Regression, Support Vector Machine and Naive Bayes which have been trained on three distinct datasets. After comparing the results of the different classifiers, they found that Logistic Regression algorithms have the best performance for the L2 normalization TF-IDF with an accuracy of 95,6%.

Watanabe et al.[19]worked on solving the problem of the detection of hate speech in tweets using Unigrams and patterns extracted automatically as features to train machine learning algorithms. They combined three public datasets (two datasets from Crowdflower and one dataset from Github) into one big dataset categorized into three classes (clean, offensive and hateful). [30] applied a pre-processing process on the dataset through different steps (removing hashtags

and URLs, tokenization using NLP tasks, part of speech using Gate Twitter Pos Tagger and lemmatization using NLP tasks). The authors extracted four features (sentiment-based features, semantic features, unigram features and patterns features) from tweets in a pragmatic way, and then proceeded to optimize them by setting values to the parameters in an optimal way to enhance the performance of classification. They performed multiple experiments with Toolkit Weka containing diverse classifiers in order to evaluate the classification using the machine learning algorithm"J48graft", results show that the highest accuracy was obtained with Unigram features and Patterns features in both cases (binary classification and ternary classification).

In the following table , we will summarize the different characteristics of the methods used in the approaches of the related works. The table have (06) six columns , explained as follow:

Approach: it represents the used approach in the associated paper.

Category of the approach: it means the technique of the artificial intelligence in which the approach is classified.

Data source: designates all the datasets used in the article (inputs).

Output: indicates the final result of the approach.

Used technique: means the methods used in the approach to detect hate speech in tweets.

Supported tools: 'Yes' if the approach is implemented using a programming language and 'No' if not.

Approach	Category of the approach	Data source	Output	Used technique	Supported tools
Prashanth et al., 2021	Deep learning	(Founta et al., 2018) (Davidson et al., 2017) (Park and Fung, 2017) (Golbeck et al., 2017) 258k tweets labeled as 58,1% : None, 16,6% : Hate and 25,3% : Abusive.	Hate tweets classified and clustered into different categories.	Character embeddings. Word embeddings. DL models with text only. DL models+text+ social and cultural context.	No.
György Kovács et al., 2021	Deep learning	Hasoc 2019: a corpus was created based on social media tweets in three languages(English, German and Hindi) OLID: similar to Hasoc corpus. Hatebase: downloading labeled data.	tweets classified with a higher score by leveraging the datasets.	convolutional and recurrent layers + LSTM Transfer learning model (RoBERTa) Word embedding FastText Glove Cross Validation	Yes.

Aya Elouali Zakaria et al., 2020	Deep learning	First version: .The dataset "Religious Hate Speech Detection for Arabic Tweets" .The dataset "Italian Twitter Corpus of Hate Speech" .The research's dataset "Hate speech dataset annotated for Portuguese" .The dataset "is-hate speech-detection" .The "Automated Hate Speech Detection and the Problem of OffensiveLanguage" Second version: .The dataset "GermEval-2018 data repository" .The "IWG hate speech public" ."HateSpeech Hindi-English Code Mixed Social Media Text" + the first version	The comments are classified into hate speech and non-hate speech tweets.	Convolutional Neural Networks (CNN). .character level representation.	No.

Ziqi Zhang et al., 2018	Deep learning	WZ-L WZ-S.amt WZ-S.exp WZ-S.gb WZ-LS DT RM	Classification of the tweets.	convolutional and recurrent layers LSTM Word embedding Linear SVM model cross validation	Yes.
Mehdi and Laith 2020	Machine learning	Twitter datasets.	Hate speech tweets categorized.	Doc2vec model for features extraction. NB SVM NN	Yes.
Sean et al. , 2019	Machine learning	Stormfront. TRAC. HatEval. HatebaseTwitter.	Meta-classifier to detect hate speech in tweets.	case-folding. tokenization. punctuation removal. Extracting words TF-IDF from unigram to 5 gram. Extracting characters N-gram counts from unigram to 5 gram. Multiple-view support vector machine classifier. considering accuracy and F1 macro for evaluation.	No.

Aditya et al., 2018	Machine learning	Two datasets available on 'Crowdflower' labeled as 'Hateful', 'Offensive' and 'Clean'. A dataset available on 'Github' which contains tweet-ID and class: 'Sexism', 'Racism' and 'Neither'.	Classified tweets into three categories: 'Hatful', 'Offensive', 'Clean'.	N-gram features with TF-IDF normalization(L1 and L2). Naive bayes. Logistic Regression. Support Vector Machine.	Yes.
HAJIME WATANABE et al., 2018	machine learning	Crowdflower : containing 14 000 tweets classified into 3 classes (hateful, offensive and clean) Crowdflower : tweets classified as (hateful, offensive and neither) Github : tweets classified as (sexism, racism and neither)	Binary classification of tweets into offensive and non-offensive. Ternary classification of tweets into, hateful, offensive and clean.	Unigram features Pattern features Sentiment based features Semantic features Natural language processing	Yes.

Table 3.2: Comparative table of related works

3.5 Analyze and discussion

Considering the previous works, it becomes obvious that the hate speech detection problem is considered as a supervised learning problem, with labeled datasets.
The studied works allowed us to analyze the performance of different ML and DL techniques and methods through the experiment's results by terms of precision recall and accuracy.
In general hate speech is defined as by the use of certain words, phrases that are offensive. However, not all tweets containing these words are considered as hate speech. That's why we need to take in consideration the context of the sentence.
Several approaches have been applied such as CNN, LSTM, SVM, NB, LR, TF-IDF, NLP, cross validation..., on different datasets. The major advantages of this approach are the ease of interpretation and the efficiency of results thanks to the multiple applied algorithms that permits to enhance the performance of the system evaluation.
On the other hand, these approaches have several imperfections such as the dependence of data size, these models are limited by the size of the dataset and often use features directly related to the data itself, which results in an "overfitting" to the training sample facing difficulties in the training phase.
Therefore, we will build our approach on the use of machine learning methods and the sentiment analysis techniques.

3.6 Conclusion

The works and researches established in the domain of hate speech detection in tweets are several due to the importance that it brings into the daily life of the human beings to live in peace in their surroundings.
In this chapter, we have established a state of art based on two categories : machine learning and deep learning based approaches by putting in place the most relevant works, besides, we have organized the different characteristics of the methods in an explanatory table with detailed analysis. In the next chapter, we will explain in detail our system architecture and proposed approach.

Chapter 4

Proposed approach

4.1 Introduction

Most machine learning techniques work well based on a common assumption: training data and test data are taken from the same feature space (from the same source).
The approach proposed in this thesis is composed of three principle phases:
A data preprocessing step in order to clean the text from all insignificant information. Next, the feature extraction phase, the dataset is transformed into a feature vector.
The final step is the classification of the dataset, including training a classifier with a train set and testing the resulting model with a test set.

In this chapter, we will present in detail our proposed approach that we used to solve the problem of identifying offensive tweets, as well as its different steps to carry out a sentiment analysis from comments.

4.2 Proposed approach

Our project consists of analyzing twitter comments in order to classify them in two categories: hate speech and non-hate speech. In order to achieve this task there are several steps to be carried out to obtain good results as follows: Data collection, pre-processing, vectorization and classification. Figure shows the conceptual diagram of our model, which contains the different steps of our approach as follows : Import the dataset from the Kaggle site, dividing the dataset into two classes: numeric data (label 0 or 1), textual data (tweet), preprocessing the data by removing unnecessary information, transforming the data into a numeric vector

using TF-IDF vectorizer in order to proceed to the classification of tweets using the classifiers: SVM, Naive Bayes and Logistic Regression.
Each step is detailed as follows :

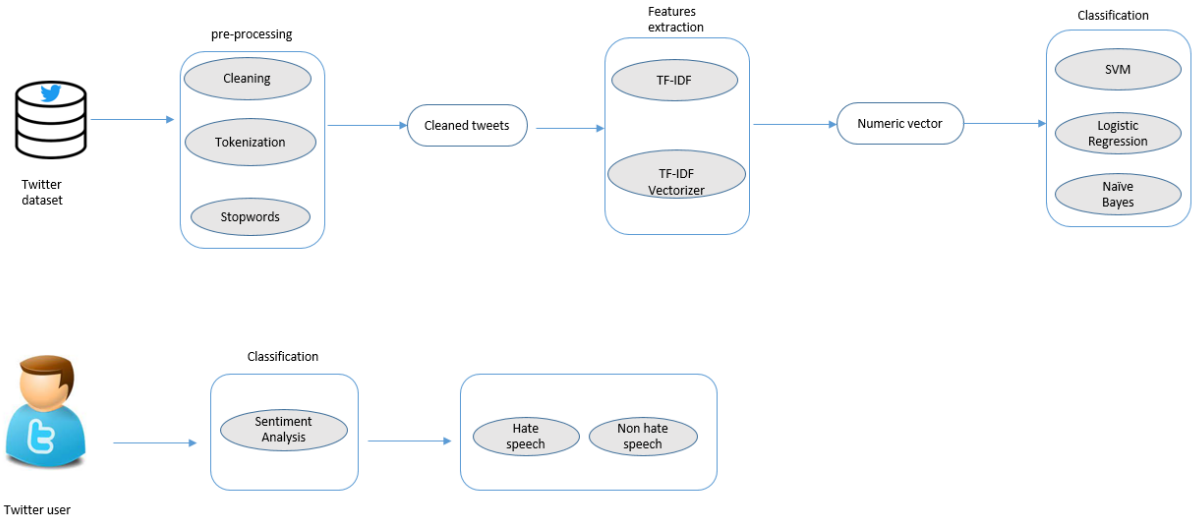

Figure 4.1: Architecture of the Proposed approach

4.2.1 Data collection

The first step in sentiment analysis is data collection. Data was taken from the Kaggle site containing [99989] tweets in english labeled as 0 for hate speech and 1 for non-hate speech[29].
The main goal of data collection is to ensure that reliable data is collected in order to facilitate data analysis.

4.2.2 Pre-processing

One of the key steps before starting the engineering process is to clean, pre-process the text where the data is prepared to become ready for analysis, by removing or modifying the data that is incorrect, incomplete, irrelevant, duplicated or incorrectly formatted.

This allows the normalization of a corpus of documents, which helps to create meaningful features and reduce noise that can be introduced due to many factors such as irrelevant symbols, special characters, XML and HTML tags, etc.
There are multiple steps in this preprocessing phase, including cleaning, tokenization and lower-casing that will be described below:

4.2.2.1 Cleaning

Is a process of pre-processing where unnecessary content such as URLs, tags, characters and punctuation is removed from text to avoid noise in the dataset [13].

Example: “ black people are slaves !! #slavery #black". “ black people are slaves".

4.2.2.2 Lower-Casing

This step is the process of standardizing the letters, where all uppercase letters are changed into lowercase for each comment.

Example: “ Ugly GIRL ". “ ugly girl ”.

4.2.2.3 Tokenization

Is a process of splitting the text into segments (tokens) where each token represents a word, in order to simplify the identification of words.
Example “ muslims are terrorists” [“muslims”, “are”, “terrorists”]

4.2.2.4 Stop-words

Words that have little or no meaning, especially when referring to construct meaningful elements from a text, such as “the”, “I”, “an”...etc known as “stop-words”, are removed from text.

Example: “ he is a black boy ” ”black boy ”

An example of the different steps of data pre-processing is illustrated in figure 4.2.

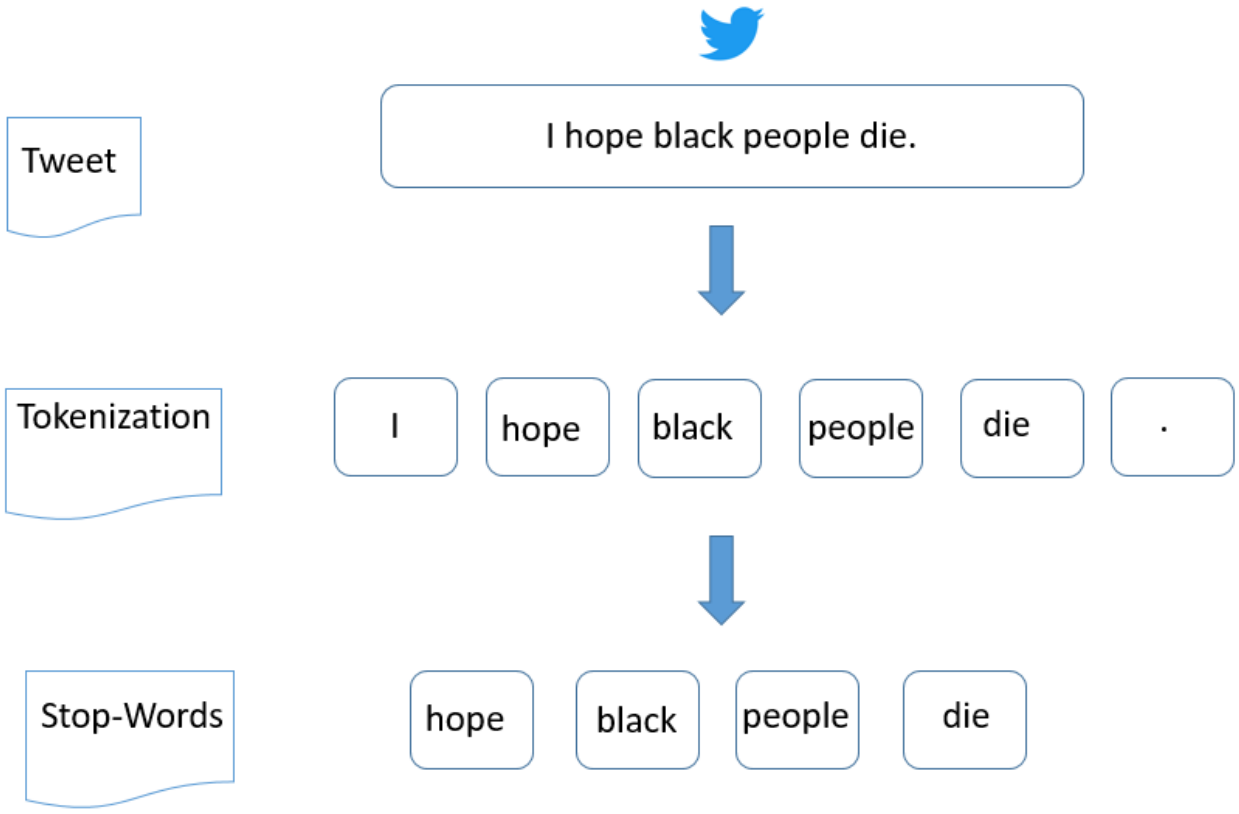

Figure 4.2: Exemple of preprocessing phase.

4.2.3 TF-IDF Vectorization

A method that permits the calculation of the frequency of occurrence of a word in a document text. We choose to work with the TF-IDF which is a technique that weights the frequency of a term (TF) and the inverse document frequency (IDF), in order to define the importance of a term.

TF: which is the correlation between the number of appearances of a word in a sentence and its length. For example, a word that occurs 4 times in a sentence of length 10 is not the same as when the sentence length is 50 words. A term with a higher TF is more important than a smaller TF.

IDF: the logarithm of the inverse of the ratio of a word appearing in a document. A term with a smaller IDF has a bigger importance.

TF-IDF: the multiplication between TF and IDF which allows to assign a weight

for each word of the corpus, resulting in feature vectors used by the classifiers[8].

4.2.4 Classification

After the preprocessing phase, we divided the dataset using the train_test_split method into training data for the training model and test data in order to evaluate the performance of the classifiers.

4.2.4.1 SVM Classifier

In machine learning, SVMs are supervised learning models capable of performing classification, regression and detection as well. It is a discriminative classifier that is formally designed by a separating hyperplane selected by calculating the maximum margin possible between the support vectors in the given labeled data (surprised training).
In this algorithm, the data is represented as points in space that are mapped so that the points of different categories are separated by a gap as wide as possible.
In addition to performing a linear classification, SVMs can perform nonlinear classification as well,by mapping implicitly their inputs into high-dimensional feature spaces.
SVM kernels are used to add more dimension to low dimensional spaces in order to make the segregation of the given data easier, in our case we will be using the linear kernel since the detection of hate speech is a linear classification.
Labeled data as hate speech and non hate speech is given as an input for the training model, SVM learning algorithm builds a model that permits to classify new inputs (the test set) to one of the categories.
SVM models use a subset of training points in their decision function that makes their memory efficient[28].

4.2.4.2 Logistic Regression Classifier

A specific type of generalized linear models, used for binary classification problems as it is in our case (hate speech or non-hate speech).
Contrary to the linear regression that is sensitive to imbalanced data, logistic regression is more suitable for classification problems, it is one of the most efficient methods for the representation of linearly separable data.
In order to draw a hypothesis function to represent the binary data, we need to use a sigmoid function shaped as the letter S with two margins on the top and bottom,

this function maps the values between 0 and 1 into the two margins and labels them as hate speech and non hate speech for our case[17].

4.2.4.3 Naive Bayes Classifier

The Naïve Bayes algorithm is a machine learning that uses probability calculations according to the concept of the Bayesian approach. The concept of this approach is to combine the conditional probability and the prior probability following the Bayes theorem in order to calculate the probability of classifying samples to a certain category (calculating the probability of a tweet belonging to one of the classes).
The term 'naive' refers to the supposition that all features are conditionally independent given the value of the class variable.
This assumption of independence rarely remains correct in real-world applications, thus the description of the algorithm as naive, but it tends to be efficient and learn quickly in various supervised classification problems. This "naivety" permits the algorithm to easily classify the dataset without using complex plans in order to realize an iterative parameter estimation[20].

4.2.5 Sentiment Analysis

Is a process of computationally identifying and categorizing opinions from a piece of text, and determining whether the writer's attitude towards a particular topic is positive, negative or neutral. In order to classify a tweet for exemple:
"i love Algerian culture"

we follow a set of steps, first we apply the tokenization on the text in order to divide the statement into different sets of words as follows:
i, love, Algerian, culture.

Then we proceed to data cleaning, by removing all special characters which do not add any value to the analytics parts: love, Algerian, culture

Finally, classifying the tweet by attributing a score for each word (+1 for positive, -1 for negative and 0 for neutral):

love = +1
Algerian = 0
culture = 0

we combine the statemements +1+0+0 = +1, in order to calculate the polarity of the text. Since the total score equals to +1, the tweet is classified as positive.

In order to determine the polarity (expressing if the tweet is hate speech, non-hate speech or neutral) and the subjectivity (expressing personal feelings view or beliefs), we use Textblob a python library for processing textual data; it will allow us to perform common NLP tasks for extraction sentiment analysis classification[25].

The figure 4.3 illustrates the sentiment analysis classification.

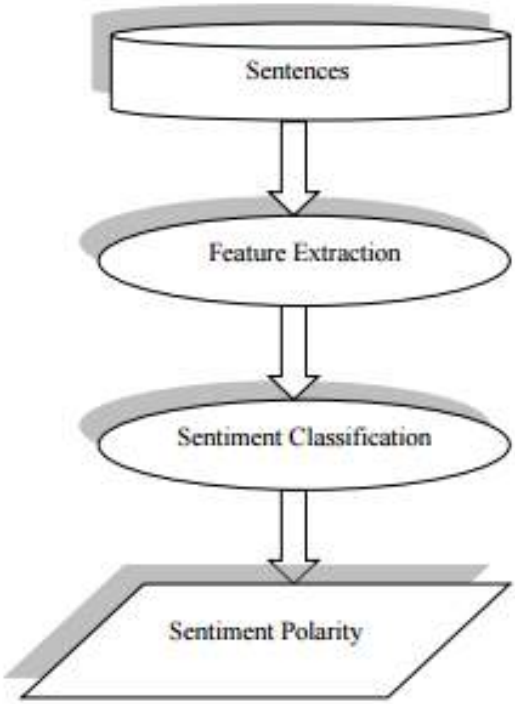

Figure 4.3: Schema illustrating sentiment analysis steps.

4.3 Conclusion

In this chapter, we started by presenting our assumptions and how our approach will solve the problem of hate speech detection in tweets using the different methods and techniques of machine learning.Then we saw the overall architecture of our pipeline,and the role of each unit in it. In addition, we presented the three classifiers SVM, Logistic Regression and Naive Bayes that we use in our approach. Finally, we ended by explaining the sentiment analysis technique used to classify

the text as positive, negative or neutral.
In the next chapter, we will proceed to explain all the aspects related to the implementation of our approach.

Chapter 5

Implementation

5.1 Introduction

In order to detect hate speech in tweets, we adopted the technique of sentiment analysis of english comments. The process of the classification will be performed on the total twitter dataset to extract the sentiment developed so it can be classified as 'Hate tweet' and 'Non-hate tweet'.

In this chapter, we will describe the dataset used to train the model, present tools used such as platforms, programming language and classification metric to evaluate the performance of our model.

In the end, we will explain and discuss the results of our experiments to evaluate the system performance.

5.2 Dataset description

The term 'Dataset' in ML represents a mechanism which can regroup, in sets, a different structure (type) of data like videos, images, text and statistics. Those data depend on several variables associated with values. Nowadays, datasets are considered as essential tools for creating models in ML.

We can divide datasets into three main categories : training, validation and test datasets. The first type of dataset is a primary tool in ML development. The exploitation of the training dataset consists of extracting and modifying necessary features to fit the models of ML before it goes to deployment.

The second type, which is the validation dataset, occurs at the end of the training step. It verifies the parameters of the model and makes modifications if necessary to have the best configuration of the model. The exploitation of data is less compared to the previous step.
Finally, the test dataset checks the performance of the system before deployment by using new values of features and predictive functions. In order to estimate the real power of the model we launch one or more tests without modifying the parameters of the model.

In our case, we have used the 'Twitter Sentiment Analysis' dataset. It can be easily downloaded from the official site of 'Kaggle'[29] in CSV form to facilitate the exploitation of the data using Python. The total size of the dataset is 8260 KB and it is a training dataset with 99989 labeled tweets.
The training dataset have three columns :

- ItemID : an integer (ID) assigned to each tweet.
- Sentiment : class label associated to the tweets
  Label 1 means that the tweet is positive (does not contain hate speech).
  Label 0 means that the tweet is negative (contains hate speech).
- SentimentText : it represents the full tweet's text with replacing mentioned user's name with @user.

The figure 5.1 indicates the distribution of sentiments in the dataset tweets with a total of 43532 tweets labeled as hate speech (0) and 56457 tweets labeled as non_hate speech (1).

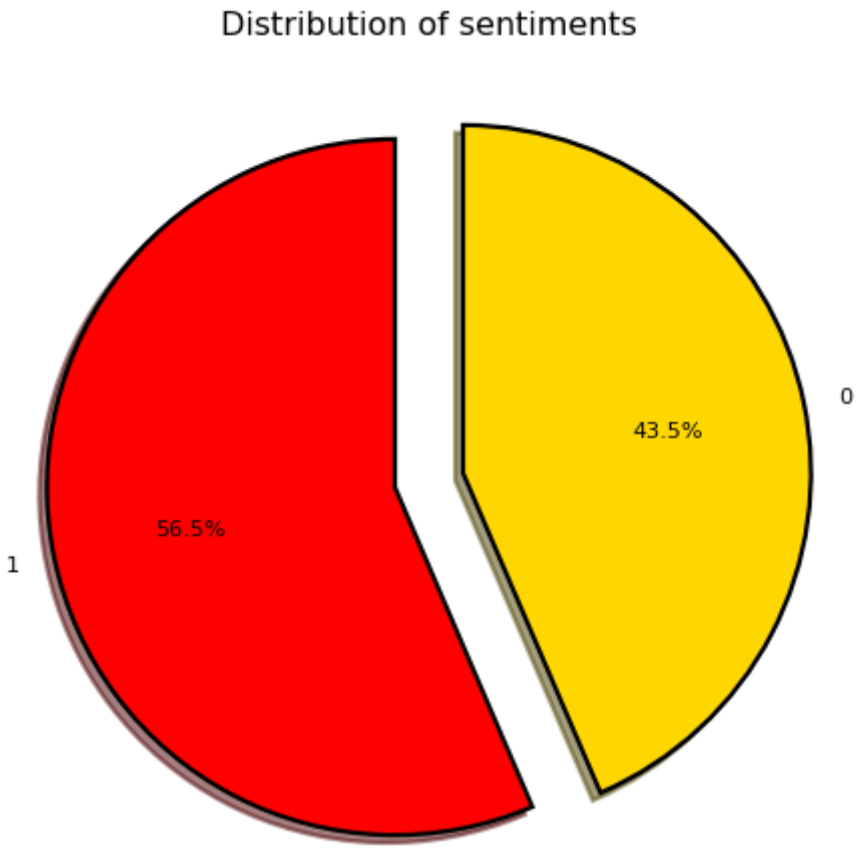

Figure 5.1: Pie chart of the dataset distibution.

the figure 5.2 shows the ten (10) first tweets in the dataset with their labels and id.

	ItemID	Sentiment	SentimentText
0	1	0	is so sad for my APL frie...
1	2	0	I missed the New Moon trail...
2	3	1	omg its already 7:30 :O
3	4	0	.. Omgaga. Im sooo im gunna CRy. I'...
4	5	0	i think mi bf is cheating on me!!! ...
5	6	0	or i just worry too much?
6	7	1	Juuuuuuuuuuuuuuuussssst Chillin!!
7	8	0	Sunny Again Work Tomorrow :-| ...
8	9	1	handed in my uniform today . i miss you ...
9	10	1	hmmmm.... i wonder how she my number @-)

Figure 5.2: Dataset simples

5.3 Tools presentation

5.3.1 Anaconda

Anaconda is an open-source platform for the python and R programming, mainly dedicated to be used in data science and ML developpement in order to simplify the steps of creating python projects.
Anaconda has over 250 packages and more than 7500 packages which can be easily installed from PyPI and managed by the package management system :conda [4].

Anaconda distribution also includes a graphical user interface (GUI) named Anaconda Navigator. This interface allows users to launch the different applications of the platform, manage conda packages besides installing, running and updating packages.

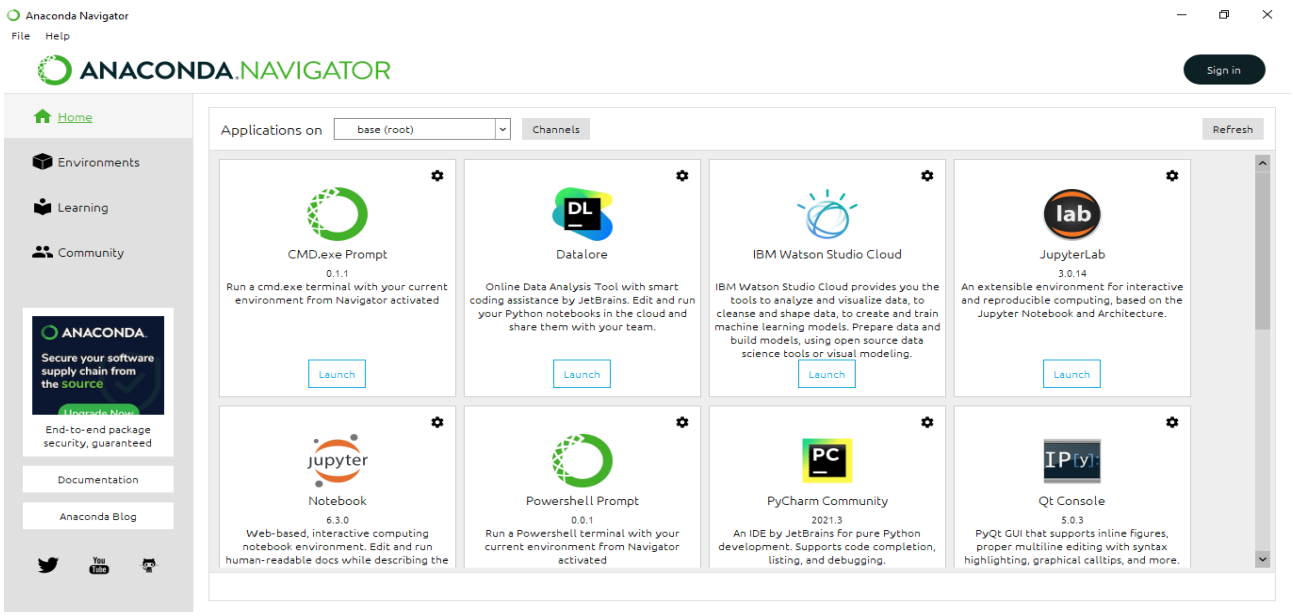

Figure 5.3: Anaconda navigator's interface

5.3.2 Jupyter notebook

Jupyter Notebook is an interactive open-source web application that supports more than 40 programmation languages (Python, R,...)[11]. The first time it deployed, its name was IPython and changed to Jupyter in the year 2014 but the notebooks (files of jupyter) have kept the format ipynb when they are saved.

Jupyter notebooks can be converted to HTML, PDF and LaTeX format by using the 'Download as ' function available in the menu of the interface.

This tool allows the users to execute pieces of code in different languages, edit documents to change the outputs to have the best configuration, and also it permits to exploit datasets in ML experiments and modelings.

5.4 Programming language ' Python'

Python is a high-level, multi-purpose and most popular powerful programming language. It was created by the developer Guido van Rossum in the year 1991[12]. This open source language stands on the philosophy ' simplicity is the best' therefore it is easy to use especially for beginners in programming. Contrary to other programming languages like C and Java, Python is an interpreted language which means that the program does not need any compilation; we directly run it to have the output of the source code.
We find python applications in several fields of artificial intelligence such as machine learning, data science and natural language processing also in almost every giant company like Facebook, Instagram and Uber.

5.4.1 Python libraries

One of the key assets of programming in python is libraries. They are a collection of pre-written codes used in order to reduce the time required to code, therefore, instead of writing the codes every single time, we can simply use the 'import' function of python to prepare and install all requirement functions and classes to use them in the programmes.
Python libraries are the most important part of the language due to the facilities and advantages it offers to the developers, especially data scientists in saving them from losing their time writing a lot of code.
Among these libraries we found :

5.4.1.1 Pandas

Pandas (for Python Data Analysis Library) is one of the most popular and widely used libraries in data science. The main objective of using pandas library

is to read data in different formats such as CSV, text, Microsoft Excel and SQL databases then convert this data structure into Dataframes object. Once the data are converted, their analysis and manipulation tasks (aggregation and merge) becomes simple by using the functions groupby, agg and merge.
Pandas is an indispensable tool for data manipulation and visualization ; it allows the management of data by adding and/or removing columns from Dataframes, beside imputing missing files. This powerful library helps data scientists to work intuitively with all kinds of data without worrying about the problems that may appear due to this difference.

5.4.1.2 Numpy

Numerical Python or Numpy is a fundamental library that is dedicated to numerical calculations with python. It performs basic and advanced operations on arrays such as linear algebra, matrix calculations, random number generation and complex calculations for exemple trigonometric (np.sin()), exponential (np.exp()) and logarithmic (np.log()) functions.
Numpy is a perfect mechanism for scientific computing and performance increasing ;it speeds up the execution time.

5.4.1.3 Scikit learn

When you are a data scientist you definitely have to deal with scikit learn library! Scikit learn is an indispensable library in data science and ML programming; it provides many functions to prepare data by centering and reducing the size of the dataset, also, it optimizes the operations in ML by selecting the most relevant variables to create powerful and strong models.
Scikit learn offers several ML algorithms to solve various problem, we can classify them into:

- Classification algorithms : SVM, K nearest neighbors, ...
- Regression : Linear Regression, Logistic regression, ...
- Clustering : K- means , ...

Besides building models, scikit learn also provides a large number of metrics which allows developers to judge the quality of the algorithms produced such as accuracy and F1 metrics.

5.4.1.4 Matplotlib

Matplotlib is a python library that's dedicated to graphics plotting and data visualization.Matplotlib library has a sub-library called ' Pyplot ' which allows us to create interfaces similar to the software MATLAB. In fact, developers prefer to use the matplotlib library because it's open source and offers more facilities than the statistical language MATLAB.
Among the functionalities that it offers , we can designate the functionality of drawing all differents type of graphics for exemple : pie charts, histograms, scatter plots also 2D and 3D diagrams,therefore, these graphics allows us to visualize all the data and establish a complete detailed analysis which contribute to build robust ML models.

5.4.1.5 Tkinter

Tkinter (or TK interface) is an open source, cross platform and a basic module of python. It is the most used tool to build Graphical User Interface (GUI) programmes and desktop applications with the programming language python.
Tkinter is used to control the TK (Tool Kit) library which can be controlled by other languages like tcl, perl, ...
On another hand, tkinter admits several widgets for example: buttons, labels and text boxes which play the role of the orchestra as they organize all the events and operations of the programme.

5.4.1.6 NLTK

The Natural Language ToolKit or commonly known as NLTK is one of the most powerful libraries in NLP treatment; it is a python package dedicated to build programs to solve NLP problems. This open source platform has been written by Steven Bird, Edward Loper and Ewan Klein[22].
NLTK library provides several text processing libraries with test datasets , also, it includes graphical demonstrations. Among NLP tasks that can be performed with NLTK, we fInd: tokenization, stemming,lemmatization, visualization and sentiment analysis.

5.5 System evaluation

In this section, we will evaluate our system and discuss the results by using classification metrics dedicated to the problem of hate speech detection and sentiment analysis in tweets.

5.5.1 Dataset

After cleaning, preprocessing and feature extraction from the dataset,we have splitted our data into x_train, x_test, y_train and y_test. After testing several methods to divide the dataset, we have found that the best technique is to consider 80% of the dataset as training data and 20% as test data because as the train dataset get bigger as the model will be performant and will detect easily the hate speech in tweets and minimizing the errors (false negative and false positive).

5.5.2 Sentiment analysis using polarity and subjectivity

The figure 5.4 below shows the sentiment analysis interface to determine the rate of the hate speech contained in tweets. The interface have a :

- **Text input field :** space dedicated to write the tweet that will be analyzed.
- **“Clean it” button :** is used to delete all text written in the text input field.
- **“Tokenization” button :** is used to separate the tweet written into a set of words.
- **“Sentiment” button :** is used to calculate the percentage of hate speech in the tweet.
- **“Analyze” button :** permits to calculate the polarity and subjectivity of the tweet.
- **"Clean result" button :** is used to delete the result of the analyze.
- **"Quit" button :** to close the interface.

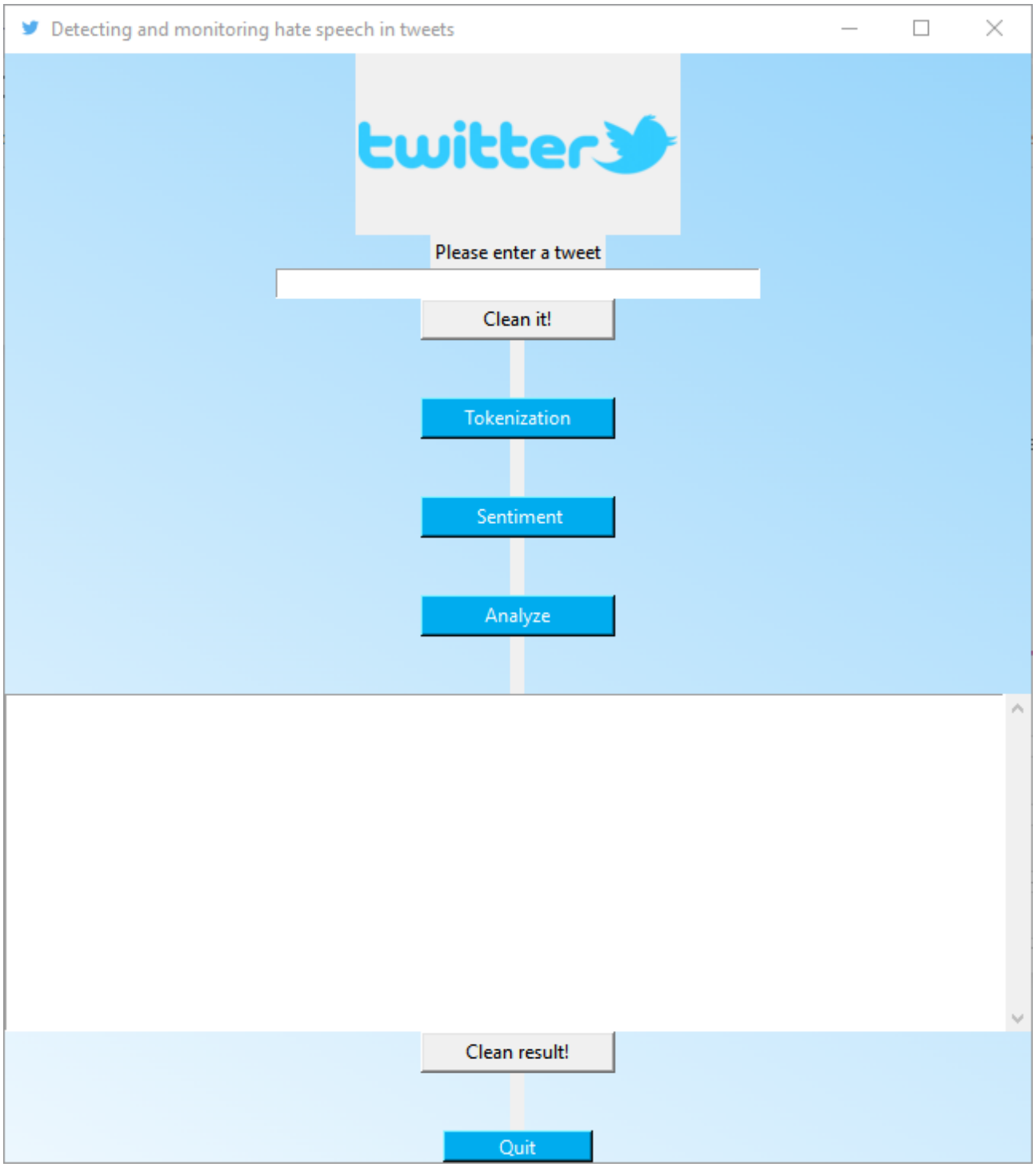

Figure 5.4: Interface of tweets sentiment analysis

the figure 5.5 shows an example of tweet sentiment analysis.

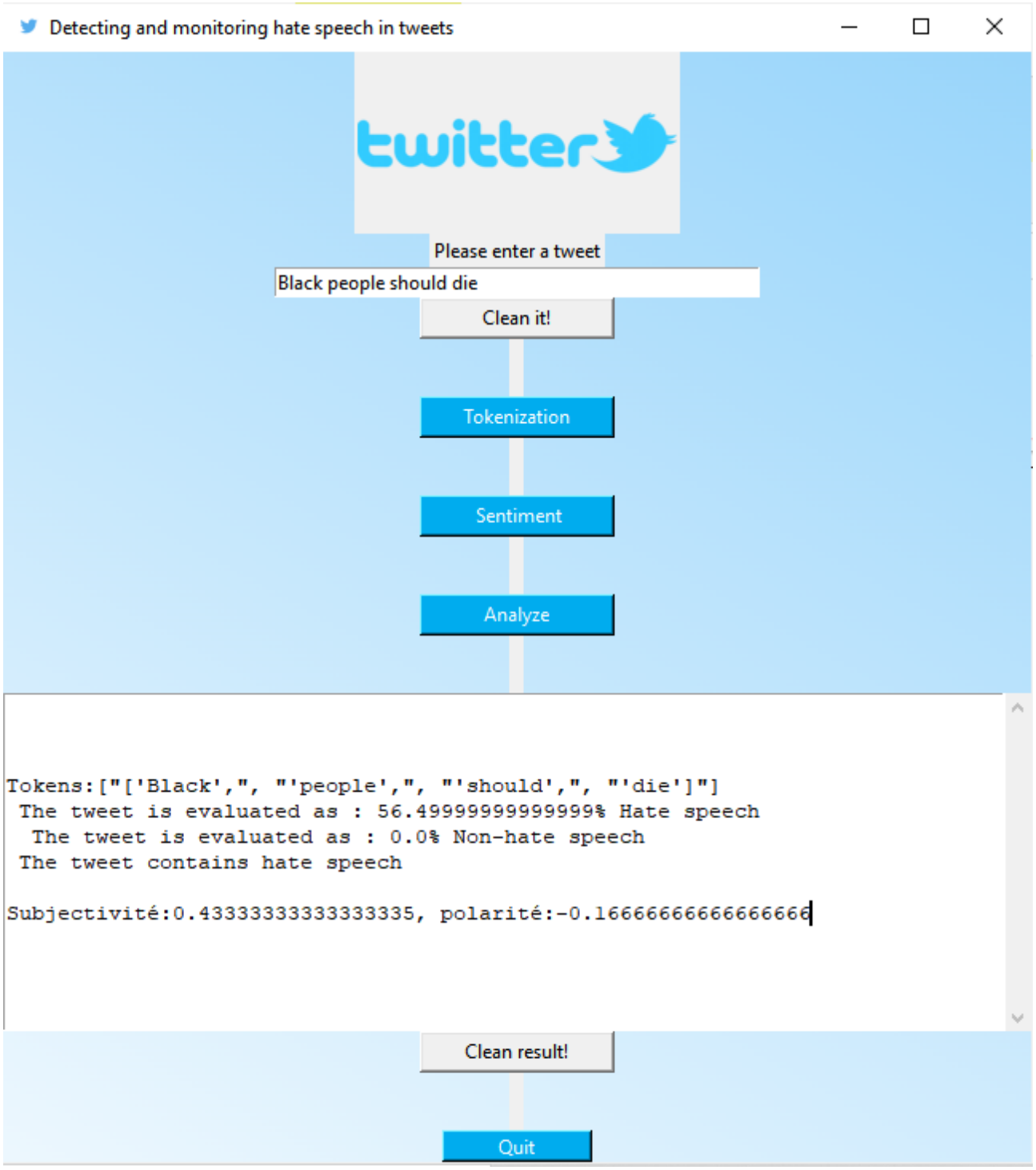

Figure 5.5: Interface of hate speech tweet example.

5.5.3 Classification using ML algorithms

5.5.3.1 Classification metrics

- **True Positive (TP) :** is an instance that has been correctly predicted by the ML model as a positive one; for example in our case, the tweets that contain hate speech will be classified in hate class.
- **True Negative (TN) :** is an instance that has been correctly predicted by the ML model as a negative one; in our case a non hateful comment will be classified as non hateful.
- **False Positive (FP) :** is an outcome classified incorrectly by the model; instead of classifying it as negative, the instance is predicted as positive one.
- **False Negative (FN):** is the case when the positive class is predicted as negative one. The model classifies a hateful comment as non hateful.
- **Accuracy :** is a metric dedicated to evaluate the classification of models ; the outcome is a percentage of the correct predictions of the model, it is calculated by dividing the number of correct predictions (TP, TN) by the number of the total predictions(TP, TN, FP and FN).
- **Precision :** is a ML indicator that evaluates a model's performance. It is the division of the relevant elements (TP) by the total number of positive predictions (TP, FP).
- **Recall :** is a measure to check out the performance of ML algorithms by dividing the true positives (TP) by the total number of the instances that should have been predicted as positive.
- **F1 score:** is a combination of the two metrics : precision and recall of the model. Generally, we consider F1 score more than accuracy as an evaluation metric especially when the dataset is imbalanced (the classes are not equal).
- **Receiver Operating Characteristic curve (ROC) :** it is a graphical representation used to visualize the performance of classification models by considering the true positive rate and false positive rate.
- **Area Under Curve (AUC) :** it is a measurement of the degree of distinguishing between two classes in the ROC curve. In order to build a smart model, the AUC score must be superior to 0.5 (> 0.5).

The table above shows the different results obtained by making several experiments on the dataset to capture the performance of each ML model in detecting hate speech in tweets and analyzing sentiments containing it.
The results are classified in the table by considering precision, recall and F1 Score for each sentiment of the ML model.

	Sentiment	Precision	Recall	F1 score
LR	negative	0.76	0.64	0.69
	positive	0.75	0.85	0.79
NB	negative	0.77	0.56	0.65
	positive	0.72	0.87	0.79
SVM	negative	0.73	0.69	0.69
	positive	0.75	0.81	0.78

Table 5.1: Comparative table of the classifiers's results.

Generally, The two metrics precision and recall can't be equal all the time ;for example, when the precision is high , the recall is lower and vice versa. Therefore, we consider F1 Score as a third metric (which is a result of combining the two metrics) to evaluate the classifier performance.

So, in our case, we found that LR and SVM models are the best classifiers to detect negative sentiment in tweets with 69% of F1 Score. On another hand, the best models to detect positive sentiment in tweets are LR and NB with 79% F1 Score.

As our dataset is a balanced one, then, we can take into consideration the accuracy metric which can decide the classement of the performance of our models , for example, NB algorithm had achieved 73% of accuracy, SVM model with 74% accuracy , finally, the last model that achieved 75% accuracy.

The following schemas show the different ROC curves of the ML algorithms using a twitter sentiment analysis dataset to detect the hate speech in tweets.

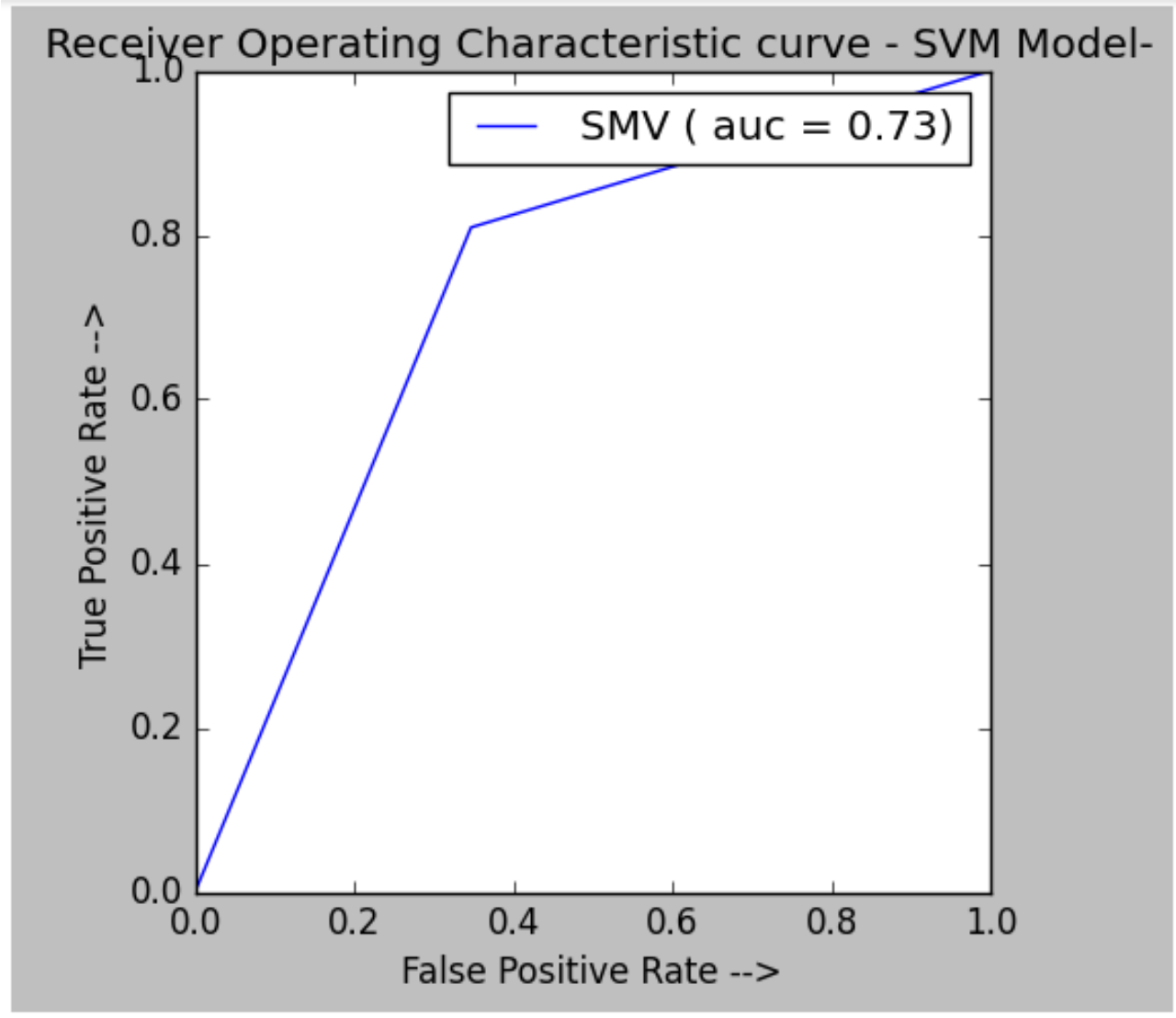

Figure 5.6: ROC curve of SVM classifier.

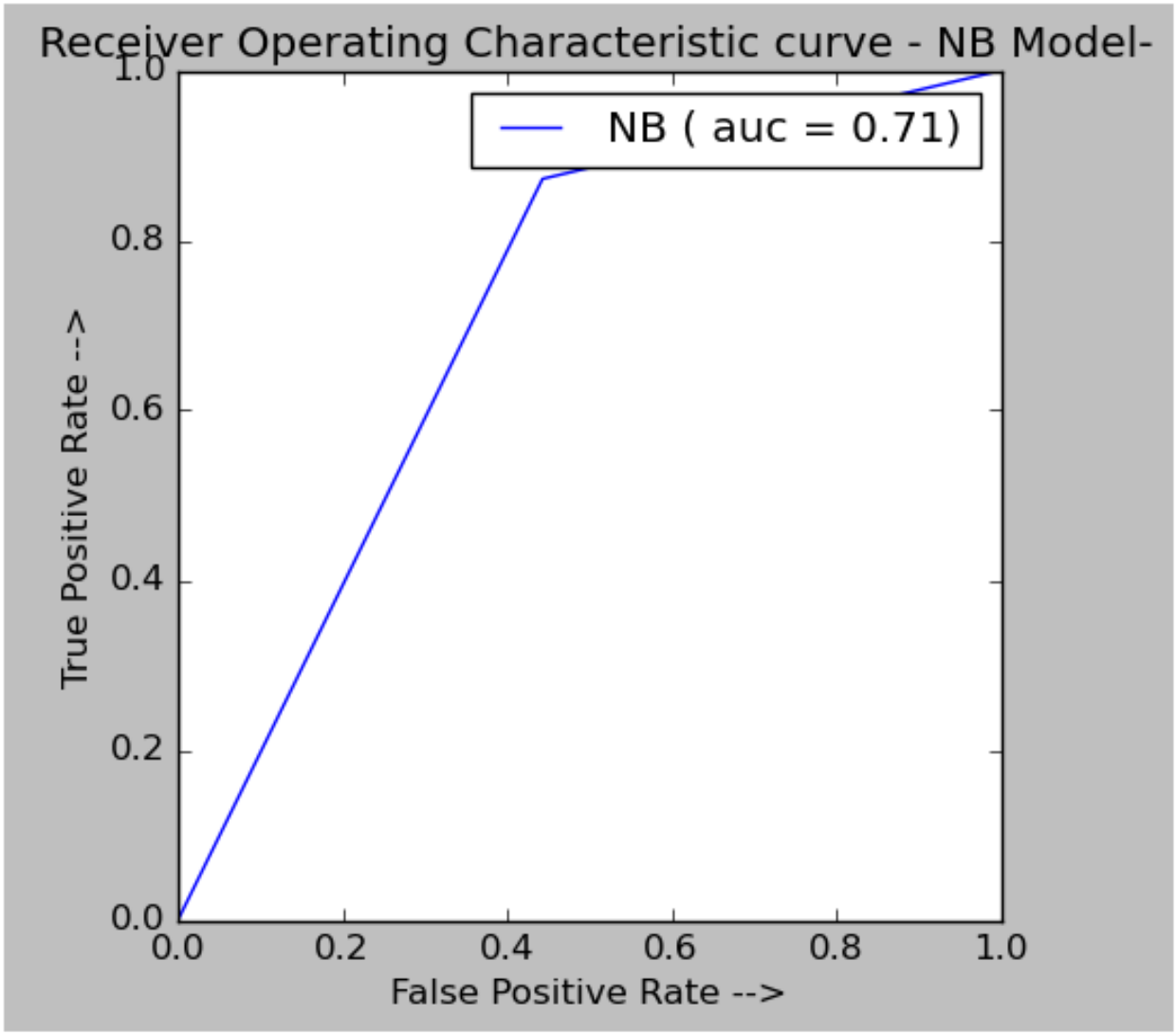

Figure 5.7: ROC curve of NB classifier.

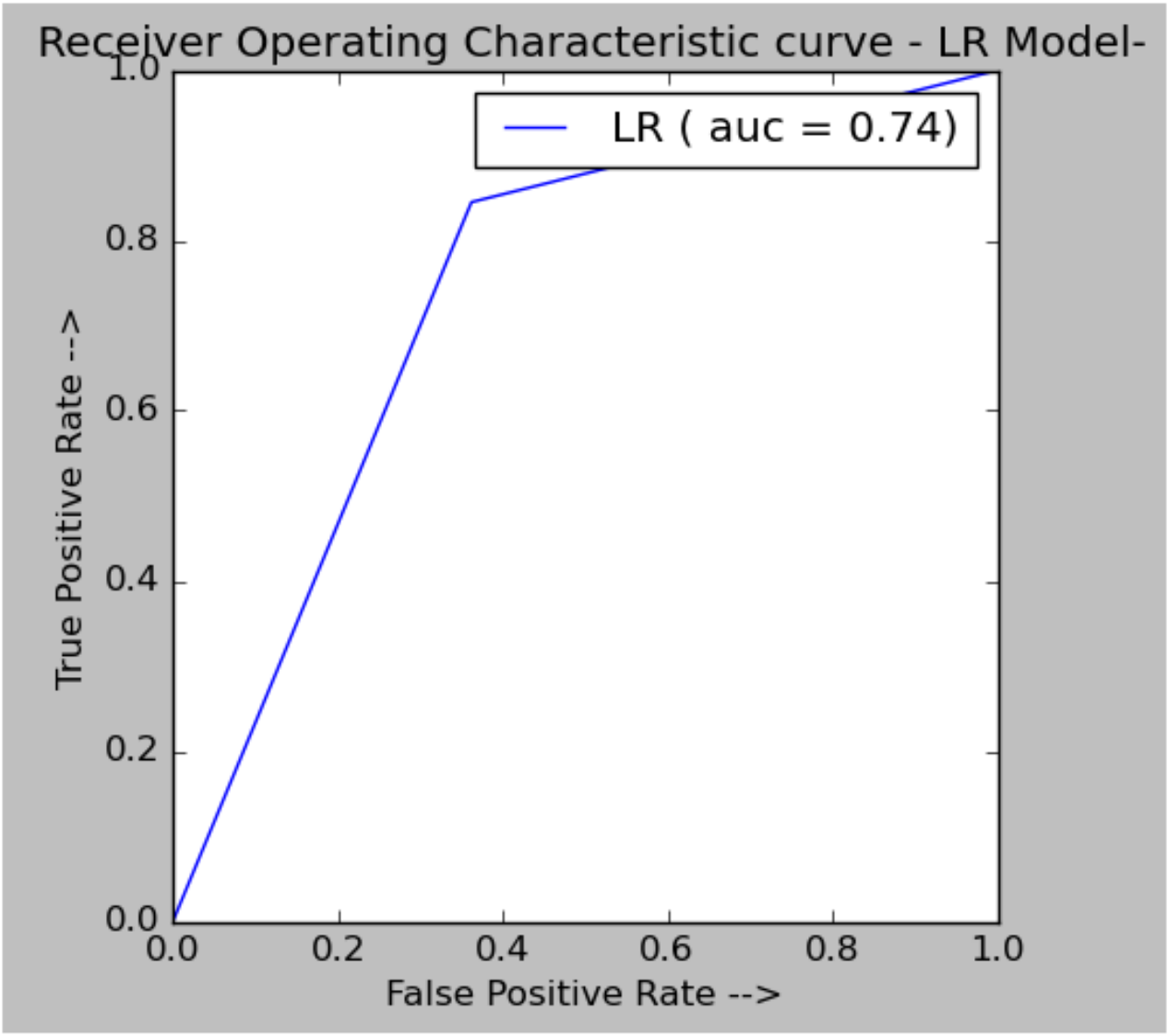

Figure 5.8: ROC curve of NB classifier.

As we notice in the diagrams of the ROC curve above, all the models are crossing the diagonal line which means that we don't have any dumb models. The ROC curve of the SVM model covers 73% of the area above the diagonal line of the diagram. NB algorithm has a score of AUC of 71% and LR algorithm which outperformed other models by making 74% of AUC.

The following figure 5.9 illustrates a ROC curve which compare the performance of the three ML algorithms.

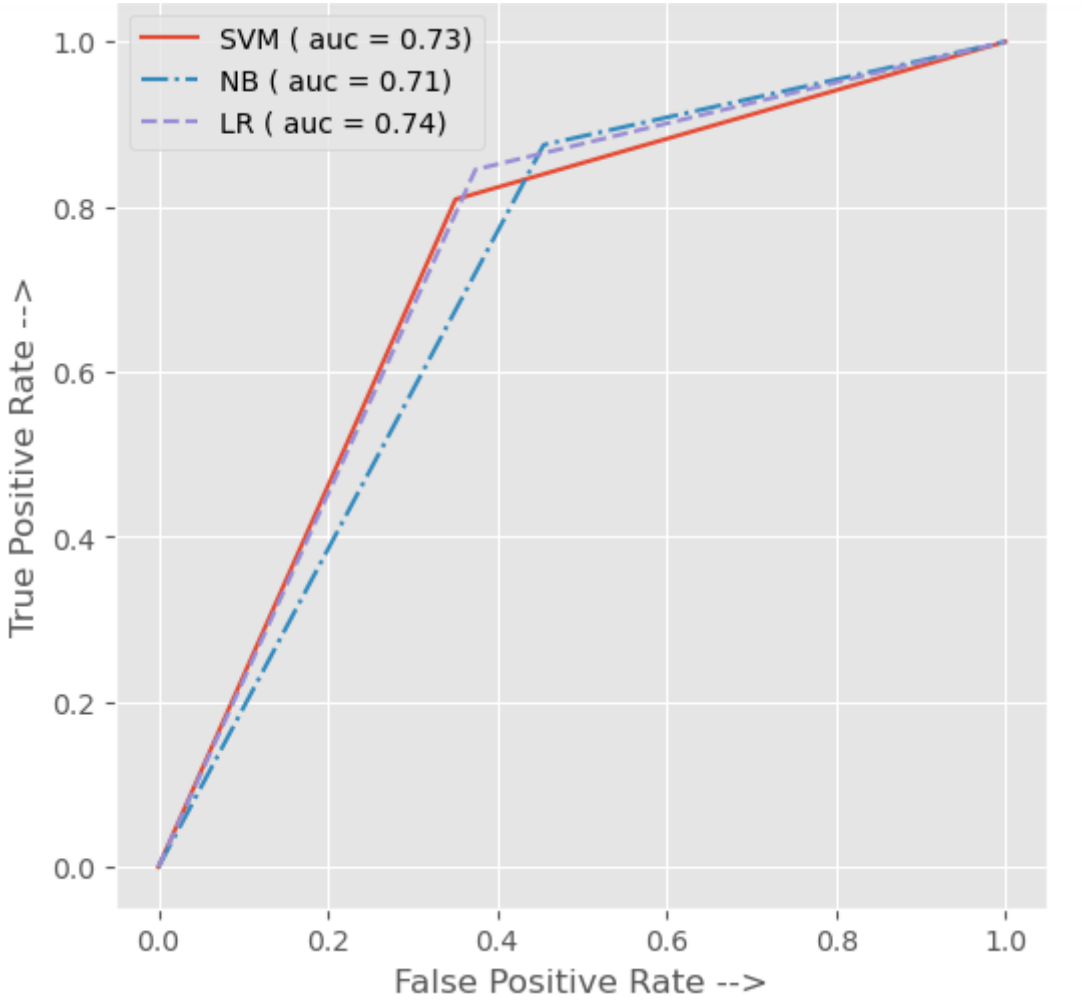

Figure 5.9: Comparasion of the three classifiers with ROC curve.

The comparaison between the three classifiers confirms the efficiency of using balanced data (no huge difference between the size of hate speech category and non-hate speech).
We notice that the ROC curves of the three models are nearlly the same, as showed in the figure, the LR classifier slightly surpasses the two other curves with an AUC score of 74% compared to the SVM and NB scores (73% and 71%, respectively.)

We conclude that LR model permits to obtain better results in the detection of hate speech in tweets.

5.6 Conclusion

In this chapter, we have discussed the main steps in our implementation. In the first place, we presented the dataset that has been used to train and test the ML models. Besides, we have detailed the principal characteristics of the work environment and the tools used to build a strong system able to detect hate speech in tweets and classify the sentiment analyzed into hate or non-hate using three ML algorithms : LR, NB and SVM. In order to evaluate the performance of our models, we have adopted several classification metrics such as accuracy, F1 score and ROC curve to visualize the results and discuss the performance of the ML algorithms.

In the next chapter, we will present the general conclusion of the thesis by discussing our contribution in the problem of hate speech detection in tweets and the future perspective.

Chapter 6

General Conclusion

Due to the great amount of information circulating in the social media platforms, the supervision of users' interaction via tweets is becoming more difficult. Thus, an automated solution is needed in order to identify and classify hate speech in comments.

Our proposed approach mainly relies on machine learning algorithms: SVM, Logistic Regression, Naïve Bayes and sentiment analysis classification that has gained the attention of multiple researchers for the automatic text processing using the Kaggle dataset.

Our thesis is constituted of six chapters organized as follows:

The first chapter consists of a general introduction to our project, we exposed as well the problematic of detecting hate speech in tweets and concluded the chapter with the organization of this thesis.

In the second chapter, we presented the different definitions related to the domain, that is the definition of hate speech , its types, the different methods of artificial intelligence(machine learning and deep learning techniques) used for text classification and the sentiment analysis approach.

In the third chapter, we elaborated the state of art of existing works categorized by the methods used (ML and DL approaches), we summarized in a table the different studies works, then we proceeded to an analysis comparison between the approaches.

In the fourth chapter, we presented in detail our proposed approach for solving the problem of detecting hate speech in comments, as well as its different phases to carry out an analysis of sentiments from the comments.

In the fifth chapter, we discussed the multiple points related to the implementation of our developed approach, the used dataset, the different softwares and tools chosen to implement our approach. Finally, we concluded the chapter with the

results of the experiments, followed by a discussion of the performance of the model.

The experiment's results illustrate the efficiency of the preformed methods in identifying offensive comments and classifying them as hate speech or non hate speech. The results of the experiment show that the best performance is achieved when a logistic regression model is used with an accuracy of 74%.

However, the detection of abusive text remains a hard task to complete because of the complexity of the human language, thus the identification of negative text should be considered rather as a regression problem and therefore calculate the probability of a tweet containing hate speech instead of just classifying the text into hate speech or non hate speech (binary classification).

The realization of this project has successfully been conducted to achieve satisfactory results with the proposed techniques, and allowed us to gain more expertise in the domain of hate speech.

Bibliography

Al-Hassan.A, A.-D. (2019). Detection of hate speech in social networks: A survey on multilingual corpus. In *In6th international conference on computer science and information technology* (p. 18).

Alshalan.R, A. A.-B. A., Al-Khalifa.H. (2020). Detection of hate speech in covid 19 related tweets in the arab region: Deep learning and topic modeling approach. *Journal of medical Internet research*, 12.

Alzubi.J, N., & Kumar.A. (2018). Machine learning from theory to algorithms: an overview. In *In journal of physics: conference series.*

Anaconda. (n.d.). Retrieved 2022-06-11, from `https://www.dominodatalab.com/data-science-dictionary/anaconda`

Dohmen.C. (2019). *Detecting hate speech in social media - a machine learning approach* (Bachelor thesis in Science Applied Computer Science). the Hamburg University of Applied Sciences.

Elouali1.A, E., Elberrichi1.Z. (2019). Hate speech detection on multilingual twitter using convolutional neural networks. *Intillegence Artificial Revue*, 8.

Gaydhani.A, K. B., Doma.V. (2018). Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv*, 5.

Gebre.B.G, W. H., Zampieri.M. (2013). Improving native language identification with tf-idf weighting. In *In proceedings of the eighth workshop on innovative use of nlp for building educational applications* (p. 216-223).

A gentle introduction to pooling layers for convolutional neural networks. (2019). Retrieved 2022-06-26, from `https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/`

Hate speech. (2022). Retrieved 2022-06-26, from `https://plato.stanford.edu/entries/hate-speech/`

Jupyter notebook. (n.d.-a). Retrieved 2022-06-11, from https://www.dominodatalab.com/data-science-dictionary/jupyter-notebook

Jupyter notebook. (n.d.-b). Retrieved 2022-06-11, from https://www.geeksforgeeks.org/python-language-introduction/?ref=lbp

Kerim.Y. (2019). *Développement d'une plateforme de détection des publications à caractère haineux dans les réseaux sociaux.* (Master thesis in Science Applied Computer Science). The superior computing National School (ESI).

Kovács.G, S., Alonso1.P. (2021). Challenges of hate speech detection in social media. *Computer Science*, 15.

Leskovec.J, U., Rajaraman.A. (2014). *Mining of massive data sets.*

Logistic regression. (2022). Retrieved 2022-05-31, from https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

Logistic regression for machine learning and classification. (n.d.). Retrieved 2022-06-20, from https://kambria.io/blog/logistic-regression-for-machine-learning/

MacAvaney.S, Y. R. G.-. F., Yao.H.R. (2018). Hate speech detection: Challenges and solutions. *PloS one*, 16.

Manaa1.M.E, A. (2020). Leveraging social data for hate speech classification. In *2nd international scientific conference of al-ayen university* (p. 13).

Naive bayes classifier. (n.d.). Retrieved 2022-06-20, from https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

natural language processing (nlp). (n.d.). Retrieved 2022-06-01, from https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP#:~:text=Natural%20language%20processing%20(NLP)

Natural language toolkit. (n.d.). Retrieved 2022-06-17, from https://www.techopedia.com/definition/30343/natural-language-toolkit-nltk

Naïve bayes classifier algorithm. (n.d.). Retrieved 2022-05-30, from https://www.javatpoint.com/machine-learning-naive-bayes-classifier

Neural network. (n.d.). Retrieved 2022-05-31, from https://deepai.org/machine-learning-glossary-and-terms/neural-network

A quick guide to sentiment analysis / sentiment analysis in python using textblob. (n.d.). Retrieved 2022-06-20, from https://www.youtube.com/watch?v=O_B7XLfx0ic&ab_channel=edureka%21

R.Yamashita, R. K., M.Nishio. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 611–629.

Support vector machine algorithm. (2021). Retrieved 2022-05-30, from https://www.geeksforgeeks.org/support-vector-machine-algorithm/#:~:text=Support

Support vector machine in python / machine learning in python tutorial / python training. (n.d.). Retrieved 2022-06-20, from https://www.youtube.com/watch?v=2v430er9hkI&ab_channel=edureka%21

Twitter sentiment analysis. (n.d.). Retrieved 2022-06-15, from https://www.kaggle.com/datasets/imrandude/twitter-sentiment-analysis

Vijayaraghavan.P, R., Larochelle.H. (2021). Interpretable multi-modal hate speech detection. *arxiv*, 5.

WATANABE.H, O., BOUAZIZI.M. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 11.

Webb.G.I, M., Keogh.E. (2010). Using the naive bayes as a discriminative classifier. *Encyclopedia of machine learning*, 3.

Wulczyn.E, D., Thain.N. (2017). Ex machina: Personal attacks seen at scale. In *In proceedings of the 26th international conference on world wide web* (p. 1391–1399).

Zhang.Z, T., Robinson.D. (2018). Hate speech detection using a convolution-lstm based deep neural network. *Springer*, 10.

Abstract

Social media is one of the most popular means of communication used today such as Facebook, Instagram, YouTube and Twitter. With the rise of modern and social media use, online interactions have become much more difficult to supervise, in particular abusive comments containing hate speech. Hate speech can be a motive for "cyber conflict" which can influence both individuals and communities. Therefore, social media services are aiming to limit these sorts of offensive comments without violating the right to freedom of expression. However, identifying if a text contains hate speech or not is still a challenging task for both machines and humans due to the complexity of human language. In this paper, we will present a background on hate speech and its related detection approaches. Furthermore, we present our work on detecting and monitoring hate speech-language in tweets using machine learning methods: SVM, Logistic Regression, Naive Bayes and sentiment analysis classification. We explain in detail our proposed approach to identify and classify abusive text in Kaggle dataset tweets into two categories (hate speech and non-hate speech), and evaluate the performance of the applied models. Our results showed that the method that permits to obtain the best scores is logistic regression with an accuracy of 74%.

Keywords

Hate speech, Machine Learning, Sentiment Analysis, SVM, NB, LR, Classification, TF-IDF, Detection.