République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Master's report

In computer science

Specialty: *GSI*

Theme

Predicting Depression Levels Using Social Networking

Guided by

- Dr.AMAD Mourad

realised by

- Ms.Aissaoui Sonia
- Mr.Belhadjer Samir

2019/2020

Thanks

Throughout the process of preparing and completing this project, we have received many assistance and guidance from various parties.

Without these individuals who are willing to share their experiences and time to give us a helping hand, we may not have completed the project on time or in a better quality.

Thus, in this section, we would like to express our gratitude to all of these individuals who had supported us.

We would like to thank our supervisor, Mr AMAD Mourad who has guided us on how to prepare high-quality documentation and presentation for the project.

With his constructive comments and suggestions, we could clearly see where our mistakes are and improve ourselves further throughout the processes in developing our work.

dedications

I dedicate this work :

To the one who transmitted life, love and courage to my dear mother all my joys,my love and my gratitude.

To my father who has supported me mentally and financially.

To my sisters :yassmina,samira,lydia.

To my brother Rayan.

To my partner in this work :Sonia

To all my friends especially Molkhir and Siham .

Belhadjer Samir

Dedications

I dedicate this work :

To my one and only, my beloved mother for her prayers for me.

To my hero,the man of my life,my father for his assistance and encouragement.

To my pumpkin twin LOUBNA , my soulmate KATIA ,my beloved WIDAD and my unique Milina

To my handsome brother ZINEDINE.

To my partner SAMIR,for his great work and patience .

To all my friends, in particular Molkhir and Siham.

To all my familly members.

Aissaoui Sonia

Abstract

Nowadays, text processing is a major field in machine learning that deals with the interaction between computers and humans using the natural language Predicting depression levels from social media is one of the most active research areas in natural language processing and text mining. This issue has many solutions offered with different classification techniques for machine learning. We hybridized two of the most known machine learning algorithms, convolutional neural network and long short-term memory to get better results. **Key words**: convolutional neural network, long short-term memory, machine learning, text processing, social media . . .

Résumé

De nos jours, le traitement de texte est un domaine majeur de l'apprentissage automatique qui traite l'interaction entre les ordinateurs et les humains en utilisant le langage naturel. La prédiction des niveaux de dépression à partir des réseaux sociaux est l'un des domaines de recherche les plus actifs dans le traitement du langage naturel et l'exploration de texte. Ce problème est traité par différentes techniques de classification pour l'apprentissage automatique. Nous avons hybridé deux algorithmes d'apprentissage automatique les plus connus, un réseau de neurones convolutif et une mémoire à long terme à court terme pour obtenir des meilleurs résultats. **Mots clés**: réseau de neurones convolutifs, mémoire à long terme à court terme, apprentissage automatique, traitement de texte, réseaux sociaux . . .

ملخص

في الوقت الحاضر ، تعد معالجة النصوص مجالا رئيسيا في التعلم الآلي الذي يتعامل مع التفاعل بين أجهزة الكمبيوتر والبشر باستخدام اللغة الطبيعية. يعد التنبؤ بمستويات الاكتئاب من مواقع التواصل الاجتماعي أحد أكثر مجالات البحث نشاطا في معالجة اللغة الطبيعية واستخراج النصوص. عولجت هذه المشكلة بواسطة عدة تقنيات تصنيف مختلفة في مجال التعلم الآلي. قمنا بتهجين اثنين من أكثر خوارزميات التعلم الآلي شهرة ، وهما الشبكة العصبية التلافيفية والذاكرة طويلة قصيرة المدى للحصول على نتائج أفضل. **الكلمات المفتاحية**: الذاكرة طويلة المدى ، التعلم الآلي، معالجة النصوص ، مواقع التواصل الاجتماعي،الشبكة العصبية الالتفافية ،. . .

Contents

List of Figures

List of Tables

List of abreviations

AOL	America OnLine
MCIemail	Microwave Communications, Inc.
GUI	Graphical User Interface
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
NLTK	Natural Language Toolkit
API	Application Programming Interface
CNN	Convolution Neural Network
1D	One Dimension
LSTM	Long Short-Term memory
TF-IDF	term frequency–inverse document frequency
TIOBE	The Importance of Being Earnest
AT&T	American Telephone and Telegraph
WHO	World Health Organization
ADAboost	ADAptive Boosting
CES-D	Center for Epidemiologic Studies Depression
Amazon EC2	Amazon Elastic Compute Cloud
CSV	Comma-separated Values
URL	Uniform Resource Locator
CBOW	Common Bag Of Words
RNN	Reccurent Neural Network
COVID-19	Coronavirus disease

General introduction

The Web offer a prodigious world of information, it has evolved from a simple set of static pages to more web services and more complex applications.

With the advent of digital technology and smart devices, a large amount of data is generated every day, especially from social networks which allow millions of users to communicate with their friends and share information, thoughts, moods and their feelings.

All these data are extracted from social media, this gives opportunities for people working in the health sector to get insight of what might be happening at mental state of someone who posted a topic in a specific manner. With this amount data provided from social media like facebook and twitter, artificial intelligence applications are mainly used in many fields, like sentiment analysis, business analytic, marketing and mental health disorder detection.

The aim of this study is to explore of natural language processing and machine learning algorithm using an English pre-labelled tweets dataset extracted from twitter to detect depression. We present a model which is an hybridization between convolutional neural network and long short-term memory, we also implement some baseline models like tf-idf and naive Bayes classifiers and compare their result with our adopted model.

The manuscript of our work is structured as follows:

- Chapter 1: Presents generalities on social networks which introduces the main characteristics and typologies of social networks, their advantages and disadvantages, thus their current problems which influence on person community.
- Chapter 2: Is specified for depression and its language identification, related work on the detection of depression from social media posts, their methodology and their results.
- Chapter 3: Explains the necessary and detailed steps for training the hybrid convolutional neural network and long short-term memory.
- Chapter 4: Represents our implementation with a brief presentation of the used programming tools, the implementation of hybrid convolutional neural network and long short-term memory, and the evaluation of the model also presents the implementation of the tf-idf and naive bayes model, at the end it shows the comparison evaluation of those models.

Finally, we conclude this work with a summary reminder on the important points discussed in this project report with the presentation of some perspectives.

Chapter 1

Social Media

1.1 Introduction

In recent years, Internet has become even more universal due to the development of social media that is accessible to anyone and anywhere. The social media make it easier to post photos and videos, share feelings and opinions and get data and information compared to traditional media such as newspapers, television and radio because they are cheap, free and offer various services.

Social media has seen remarkable growth recently such as Facebook and Twitter which are websites that bring together individuals, companies and organizations with the opportunity to exchange information through interactions. With their ease of use and free access, they are growing in success day after day and gaining public confidence.

This chapter is devoted to present the well-known social media types, their characteristics, emphasizing the rising and the different uses of social media and its impact.

1.2 Social media

Social media has evolved from being an obscure jumble of technologies to a set of sites and services that are at the heart of contemporary culture. One of the first reactions that many people have when hearing the term social media is to ask: are not all media social? this questions have to do with another question: Are human beings always social or only if they interact with others? In sociological theory, there are different concepts of social:

1. Some sociologist say that all media are social because the aspect of society is present

in technological artefact we use [1]. For example, if you type a document offline, your activities are social: the ideas you think and write up refer to ideas of other people and what is happening in society.

2. Others say that not all media are social, but only those that support communication between humans. Communication is the act of conveying meanings from one entity or group to another through the use of mutually understood signs, symbols, and semiotic rules [2].

3. The third form of sociality is collaboration or co-operative work. As for example the co-operative editing of articles performed on Wikipedia or the joint writing of a document on Google Docs or editing code in GitHub[3].

1.3 Web 2.0 and Social Media

Social media and web 2.0 are terms for describing types of World Wide Web applications, such as blogs, social networking sites, or videos, images, file sharing platforms or wikis.

The term was invented by Darcy DiNucci and later popularized by Tim O'Reilly and Dale Dougherty at the O'Reilly Media Web 2.0[4]. The term means such network application and tools that increase our ability to share, to co-operate, with one another, and to take collective action as described in figure 1.1.

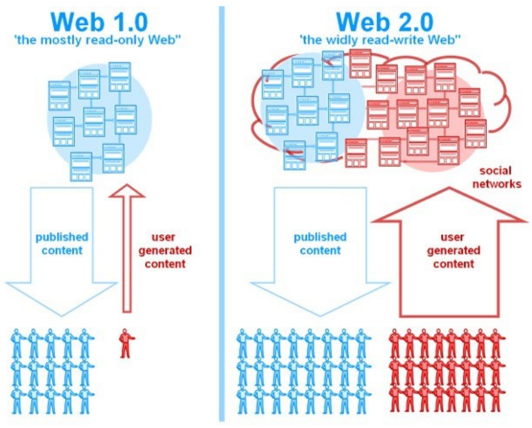


Figure 1.1: Web 1.0 and web 2.0

1.4 Type of social media

1. **Social networks:** It is one of the most well-known types of social media. Social networking is the use of online platforms to stay connected with friends, family, colleagues, customers, or clients. Defined as websites that facilitate the building of a network of contacts in order to exchange various types of content online. The most popular social networks are Facebook[1], Twitter[2], LinkedIn[3], WhatsApp[4].

2. **Media sharing networks:** Are the websites that allow us to upload our photos, videos, and audios for the purpose to be accessed anywhere in the world. Media sharing networks are the most visual type of social platforms ,while sites like Facebook and Twitter enable us to post our photos and videos but they are not considered media sharing networks because they don't focus solely on sharing visuals. The most popular media sharing networks are YouTube[5], Instagram[6], Pinterest[7].

3. **Social blogging networks:** Social blogging networks are websites that allow us to share and publish content. They help people to discover, save, share and discuss new and trending content. These socials are a hotbed of creativity and inspiration for people seeking information and ideas [5]. The most known social blogging networks are Tumblr[8], Medium[9].

4. **Discussion networks:** Discussion networks are another type of social media platform, they are focusing on discussing news, information, and opinions. Before social media networks these forums were the places where professionals and experts used to do different kind of discussions. The most popular of these forums are Reddit[10], Quora[11], StackOverFlow[12].

[1]https://web.facebook.com/.
[2]https://twitter.com/?lang=en.
[3]https://www.linkedin.com/.
[4]https://www.whatsapp.com/?lang=en.
[5]https://www.youtube.com/.
[6]https://www.instagram.com/.
[7]https://www.pinterest.com//.
[8]https://www.Tumblr.com//.
[9]https://www.Medium.com//.
[10]https://www.Reddit.com/?lang=en.
[11]https://www.Quora.com//.
[12]https://www.StackOverFlow.com/?lang=en.

5. **Reviews networks:** Review networks are the websites that allow people to find, share and review different information about people, businesses, services or brands. The most popular reviews website is Yelp where people can share their opinions and experiences with a business [6].

1.5 The rise of social media

Social media has changed the world, the vast adoption of these technology has changed our life, changed how we interact with one another and the world around us and how we can access current information. Social media becomes an integral part of the modern society.

1.5.1 Social media before 2000

Everyone knows social media and use it probably everyday too. However, it was not like this two decades ago, not many people had internet access back then.

1. It started with bulletin Board System, a text-based online community that allow users to share or exchange messages or other files on networks [7].

2. CompuServe was began in 1970s as commercial online service provider[8]. CompuServe members can share files and access news and events and they also have the possibility to join discussion forums with thousands of other members.

3. In 1993 AOL (America Online) introduced its own email addresses after being online service called GameLine as Control Video Corporation company[9]. AOL was " internet before the Internet ", it offers to subscribers the possibility to send and receive emails from users of CompuServe, MCI email, AT&T mail, AppleLink, Spring Mail, and other connected systems.

4. In 1997 the first recognizable social networking website was created named SixDegrees, it was the first manifestations of social networking website in the format seen today[10]. It has allowed users to create profile and befriend others and send messages and post a bulletin board items. SixDegrees[13] had about 3 million users registered members.

[13]http://sixdegrees.com/.

1.5.2 Social media after 2000

Social media channels had been everywhere since 2000s. They became very popular; we will mention some of those websites:

1. The revolution of social media started with MySpace website, it was the first social media to reach a million active users monthly, it achieved this milestone around 2004. MySpace users can completely customize their profiles and also embed music and videos.

2. Linkedln also was one of the first social media websites, unlike MySpace, LinkedIn users can create profiles that have a structure similar to resume with the summarization of their career, their skills and the history of their education and employment, LinkedIn allows to users to build a professional network and even find a job. LinkedIn is currently available in 24 languages and has more than 15.000 full-time employees and around 660 million members in 200 countries and regions worldwide[11].

3. In 2004 Mark Zuckerberg launched Facebook, the website became the giant and the largest social networking in the world, users in Facebook can create profile, upload photos or videos, they can also exchange messages and post content. The site has many components like posting status which enables users to alert friends to their location, new feeds and so on. From the New York Stock Exchange Wall Street in Core Facebook Vitals as reported in Q3 2019 IR Statement, Facebook has 43030 employees over 2.45 billion monthly active users and 1.62 billion people on average log onto Facebook in 2019[12].

4. In the year 2006, twitter was launched. The microblogging and social media service on which users can post and interact with messages as twitters, in additional, users can track specific topics and create dialogues of a chosen type. The total number of monthly active users in twitter is 330 million, the number of tweets send per day are 550 million and the number of twitter daily active users are 139 million [13].

   Social media actually is taking control over our lives, it becomes unavoidable that if you haven't put some form of what you've done online, then you never really did it. In the figure 1.2 [14], we display some statistics about the number of social media users.

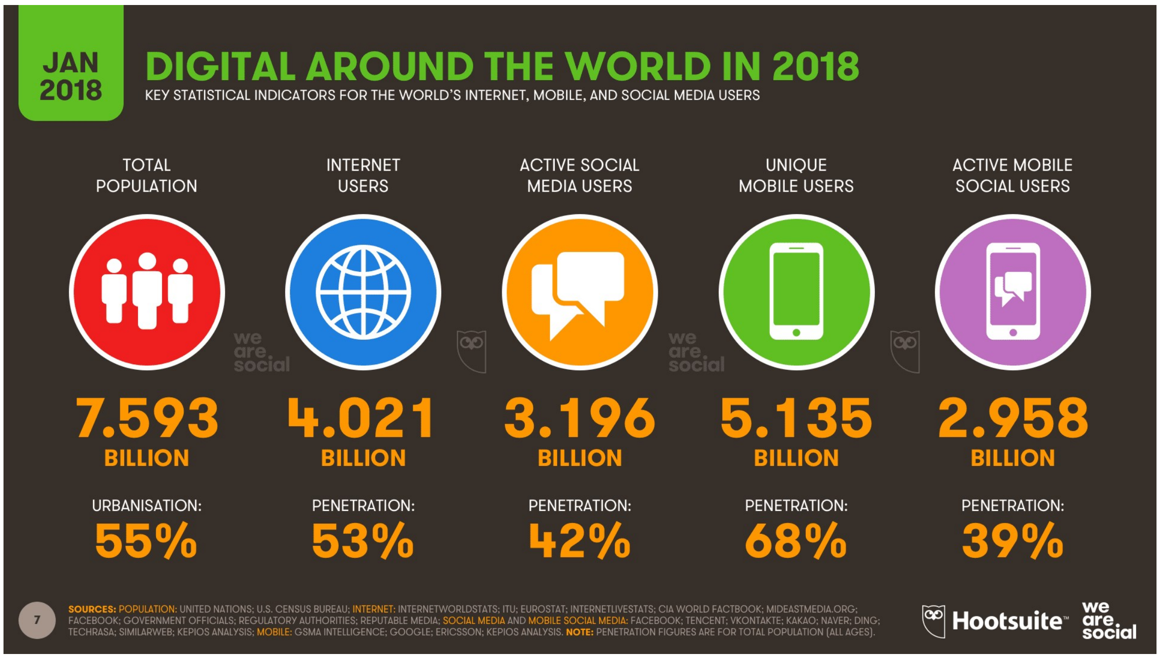


Figure 1.2: Digital around the world in 2018

1.6 Impact of social media

Social media influences on society in both positive and negative ways. We will point out some of advantages and disadvantages of social media.

Advantages

- People from anywhere can connect with anyone regardless of the location and religion.
- Enhance your knowledge about any field and get educated from others.
- Spread awareness and innovate the way people live.
- Improve business sales and reputation.
- Creating communities to discuss and share related stuffs.

Disadvantages

- Personal data and privacy can easily be hacked and shared on the internet.

- Threats, intimidation messages and rumors can be sent to the masses to create discomfort and chaos in the society.
- Waste individual time that could have been utilized by productive tasks and activities.
- Privacy can be compromised by security agencies regarding any issue discussed over the internet.
- Health issues like obesity, depression, anxiety when getting addicted.

1.7 Professional uses of social media

The rage of social media is taking over everyone nowadays. Billions of people use social media for learning, marketing, shopping and decision making. We will present its different uses:

1. **Social media in education:** Social media today made education anywhere and anytime, many studies have confirmed that the majority of students spend more time on social networking platforms which led institutions to integrate social media into the classroom curriculum to serve as useful tool to enhance students' learning. This form of learning offers many benefits to both educators and students such as encouraging the real time student engagement in courses to enhance the connection between educators and students.
2. **Social media in business:** The importance of social media in business is growing at warp speed. Marketing in social media is one of the strategies that both small and big businesses use to reach their target audience and boost sales over time. According to the statistics mentioned above, 3.484 billion social media users worldwide which is about 42% of total population, this presents a great opportunity for companies to share their content and bring new visitors to their websites or stores.
3. **Social media in government:** Social media is a powerful way for government organisations to interact with people. In those platforms, government entities can raise awareness about issues the public needs to know about and social media provides perfect avenues to remain transparent and clear with constituent as possible.

According to a survey conducted by Open the Government, more than half of voters want to see more authenticity and transparency in their government[15].

1.8 Social media constructing our reality

Although the internet may have provided an escapism from everyday life, it is mostly mimicking it. Social media users' content often highlights who they spend time with, where they go and what they do. Day after day, post after post their content forms their online identity. In this section, we present what is online identity in social media and how to optimize it and briefly speak about fake profile.

- **Online identity in social media:** Also known as internet persona, is social identity that an internet user established during his connectivity. Online identity is not the same as the real identity because the characteristics we present online differ from the ones in the physical world but it reflects kinds of interaction that often are the same from our real world interactions. The definition of identity is not only shaped by the users with what they say about themselves but also what they do ,the content, the words, the images and videos they share, and the users and groups with whom they interact.

- **Optimization of online identity in social media:** Your identity is who you are and what you do. It was creepy and dangerous to use your real identity in social media but the raise of those platforms changed that, starting from Friendster and MySpace, people began sharing their real names, pictures and interests with other people on the internet. In the following some basic tips to optimize identity in social media:

  - **Pick your username** :Your username tells people who you are ,you need to figure out who you are going to be? are you a regular person ,a brand ,a person representing a brand or a fictional character, this steps will affect who you are able to connect and interact with.
  - **Think before sharing**: Posting online is instant and permanent. Once you post, you lose all control of what will happen to it. The cyber world is the real world with the real consequences so make sure you always ask yourself

  the following question before you post a content is it true? Is it helpful? Is it necessary? Do I have a permission? Is it legal?

  – **Choose where and with whom you share**: It is necessary to distinguish between what is shared in private and public profiles, make sure you're only sharing your information with the people you want to share with, but also in direct messages to a single user who must be trustful and public messages to a group.

- **Fake profile:** A fake profile is the representation of people that does not exist in social media, it is a profile with fake details like name, and wrong images and other information, or the account that appears might be of another person, company or product.

  – **Why people fall for fake profile**: One of the biggest problems in social media is the fact that those platforms have created a lot of different avenues that certain people abuse, most frequently through the creation of fake profiles. In 2019, Facebook removed 3.2 billion fake accounts from its platform[16], likewise 9 to 15 percent of twitters accounts are fake. Fake accounts are not only in Facebook and twitter, Germany's intelligence agency says that China used fake LinkedIn profiles to spy on German politicians[17], but in the recent years social media company begun hiring more people and using artificial intelligence to detect those profiles and delete them.

Social media platforms have become a dominant source of data used by governments, corporations and academics to study human society. All of these shared updates, interactions generated by subscribers, these shared comments, images and videos, and all the activities on social media come together to form extremely useful data. This is why we chose social media data to do our research.

1.9 Conclusion

In this chapter, we have studied the different concepts necessary for understanding social medias. As known, they take up a large place in daily life especially socially and professionally which led us to be interested in detecting depression levels on social media which is the subject of the next chapter.

Chapter 2

State of art of depression prediction

2.1 Introduction

Computer science is widely used by all research fields. In mental health, machine learning can be used to develop a treatment plan and identify some key markers for mental health disorders. Depression detection with machine learning is an automatic processing task of languages and information extraction to determine them generally on two levels (depressed, not depressed).

2.2 Depression (major depressive disorder)

Depression (major depressive disorder) is a medical illness that negatively affects how a person feels, the way he thinks, moves and the way he processes and uses language. Depression causes feelings of sadness and/or a loss of interest in activities once enjoyed. Especially when long-lasting and with moderate or severe intensity, depression may become a serious health condition. It can lead to suicide. Close to 800 000 people die due to suicide every year[18].

It is a common illness worldwide, according to the World Health Organization (WHO), there are more than 264 million people affected[19].

2.2.1 Depression Language

Scientists have long tried to pin down the exact relationship between depression and language. In study recently published in Clinical Psychological Science[20]. The researchers across three studies they conducted a text analysis of 63 Internet forums (over 6,400 members) using the Linguistic Inquiry and Word Count software to examine absolutism at the linguistic level, they found that anxiety, depression, and suicidal ideation forums contained more absolutist words than control forums. In another study published in sagepub, depression influences use of first-person pronouns, whereas negative affect influences use of third-person pronouns[21].

From those and other research studies we deduce that in depressed individuals, specific patterns of a word's usage have been identified.

Negativity

New research at the 2017 Pediatric Academic Societies Meeting in San Francisco suggests that depressed person rarely used the word "depressed" when describing their feelings, but instead used other negative emotion words like "down," "stressed," or "upset"[22].

Absolutes

The study also found that people with symptoms of depression commonly write and talk in absolutist language[22]. They frequently say things like = "There is no solution", "Tomorrow doesn't exist", "I'm never going to be able to do this".

Pronouns

In the two researches we referred to below, people with depressive symptoms tend to use more first-person pronouns. As the Doctor of Philosophy Mohammed Al-Mosaiwi writes in a post for IFL Science:

"This pattern of pronoun use suggests people with depression are more focused on themselves, and less connected with others. Researchers have reported that pronouns are actually more reliable in identifying depression than negative emotion words"[23].

2.3 Machine learning

Machine learning is a field of Artificial Intelligence that makes a computer learn and act like a human, it provides to the systems the ability to automatically learn and improve from experience without being explicitly programmed. At a very high level, machine learning is the process of teaching a computer system how to make accurate predictions when fed data.

Machine learning can be divided into categories according to their purpose, the main categories are the following:

2.3.1 Supervised learning

It is referred to as learning with a teacher[33], in this type of machine learning the systems are fed with massive labelled input in the training phase, where the data set are already classified, which instruct the system what output should be obtained from each specific input value. The most used algorithms in this categorie are the following:

Linear Regression

It is a technique used in predicting, forecasting, and finding relationships between quantitative data. Linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables[34]. For example, this technique can be applied to examine the relationship between a company's advertising budget and its sales. It can be also used to determine if there is a linear relationship between a particular radiation therapy and tumor sizes.

Classifications techniques

Some practical examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification, some of widely used classifiers are :

- Logistic regression,
- Linear discriminant analysis,
- K-nearest neighbors,
- Trees,
- Neural Networks,
- Support Vector Machines.

2.3.2 Unsupervised learning

It is when the input dataset are not labelled, instead the model itself work on its own to discover patterns and information that was previously undetected. Figure 2.1 shows an example of unsupervised learning algorithm named clustering.

Clustering

The most unsupervised learning used in this type of machine learning is the technique where grouping points of data, typically, we may use clustering to discover classes within the data. For example, an unsupervised learning method can take, as input, a set of images of handwritten digits. Suppose that it finds 10 clusters of data. These clusters may correspond to the 10 distinct digits of 0 to 9, respectively. However, since the training data are not labeled, the learned model cannot tell us the semantic meaning of the clusters found[35].

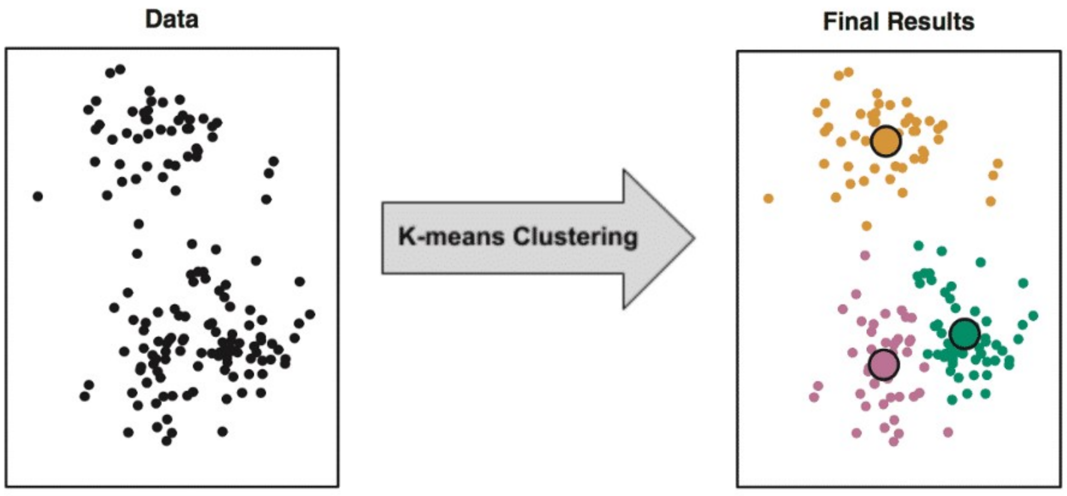


Figure 2.1: Clustering

2.3.3 Semi-supervised learning

Semi-supervised learning algorithms represent a middle ground between supervised and unsupervised algorithms, It is the combination of supervised and unsupervised learning methods. The inputs data has a few labeled data but the most of them are not labelled. For example developing a model to detect fraud for a large bank. Some frauds are known, but other instances of fraud are not, we can label the dataset with the fraud instances that we are aware of, but the rest of our data will remain unlabeled. The most used algorithms in this approach are:

- Self training,
- Graph based algorithm,
- Generative model,
- Multiview algorithm.

2.3.4 Reinforcement learning

In this type, the purpose is to train a model to make sequence of decisions, The agent learns to achieve a goal and takes actions in an environment, potentially a complexed one, (see figure 2.2.)

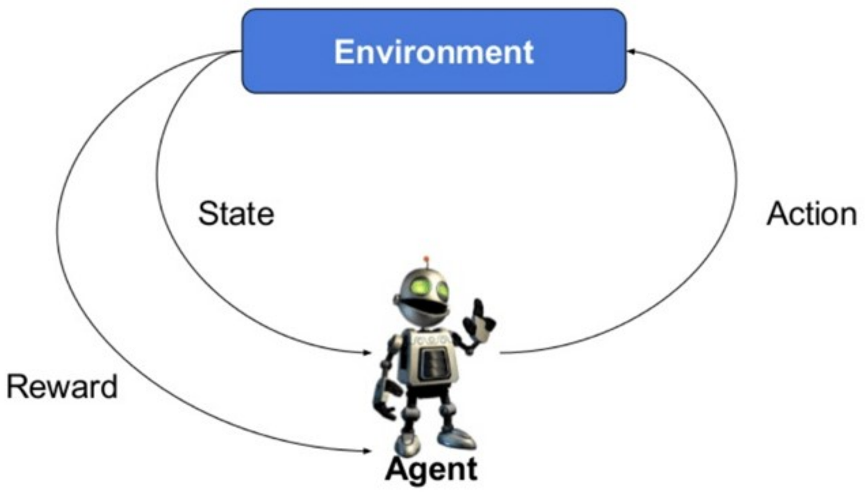


Figure 2.2: Reinforcement learning

as shown in Figure 2.2, the environment represents the outside world to the agent and an agent takes actions, the agent tries to figure out the best actions to take, he receives observations that consists a reward for his action and information about his new state. Those rewards inform him of how good or bad was the action taken, and the observation tells him what is his next state in the environment.

2.4 Machine learning and mental health

Recently, Neuroscientists and clinicians around the world are using machine learning methods to assist mental health providers, to make decision on patients historical data, as medical records, behavioral data and social media usage. One of the benefits is that machine learning helps clinicians predict who may be at risk of a particular disorder. As David Benrimoh a psychiatry resident at McGill University Machine learning really meets a specific need that we have in psychiatry and that's the need for personalization[36].

2.4.1 Identifying Biomarkers

Depression is a complex clinical entity that can pose challenges for clinicians regarding the trial and error process to achieve the right dosage of medication and adopting the right treatment plan. Individuals with major depressive disorder vary in their response to antidepressant treatment. The Machine learning algorithms could help determine key behavioral biomarkers to help mental health professionals to develop more catered treatment plans and medication dosages for their patients.

2.4.2 Predicting Crises

The important and the complex meantal health problem is the understanding when people can be more prone to crises such as panic attacks, psychosis, manic states, migraines, certain conditions such as Schizophrenia and Bipolar disorder have a higher risk of crises occurring. Mental health professionals are responsible for minimizing the risk of crises for patients. The usage of immense user data from their daily social media activities and the usage of machine learning algorithms we mentioned below, the models can predict those crises with the detection of stress, isolation, or exposure to triggers.

2.5 Related Work

Various studies have been conducted on the prediction of depression from social media posts.

2.5.1 Predicting Depression Symptoms in an Arabic Psychological Forum

In this study the researchers investigate the application of natural language processing and machine learning on Arabic text for the prediction of depression[24]. The first stage at every such study is the collection of data, the researchers have used the online forum named "Nafsany"[25] to get the data. It is an online platform where people from Arab countries publicly share their stories to obtain feedback. Table 2.1 presents the statistical characteristics of the collected corpus.

Caracteristic	**ARABDEP Corpus**
Total posts	20000
Total words	4873544
Average words per post	467
Total characters	20919136

Table 2.1: Characteristics of the corpus

The second stage in this phase is the manual annotation or labelling with help of a psychologist to distinguish between post with or without depression.

In the final stage, with the help of studies such as “Diagnostic and Statistical Manual of Mental Disorders”[26], "Quick Inventory of Depressive Symptomatology"[27], and “Patient Health Questionnaire”[28], the researchers manually annotating the data according to rules that are based on depression symptom classes identified from the studies mentioned above [26].

Prediction models Lexicon-Based Approach

1. **Construction of lexicons:** the researchers first in this approach generate a depression lexicon namely the ArabDep lexicon that collect depression-related terms

with data collected above.

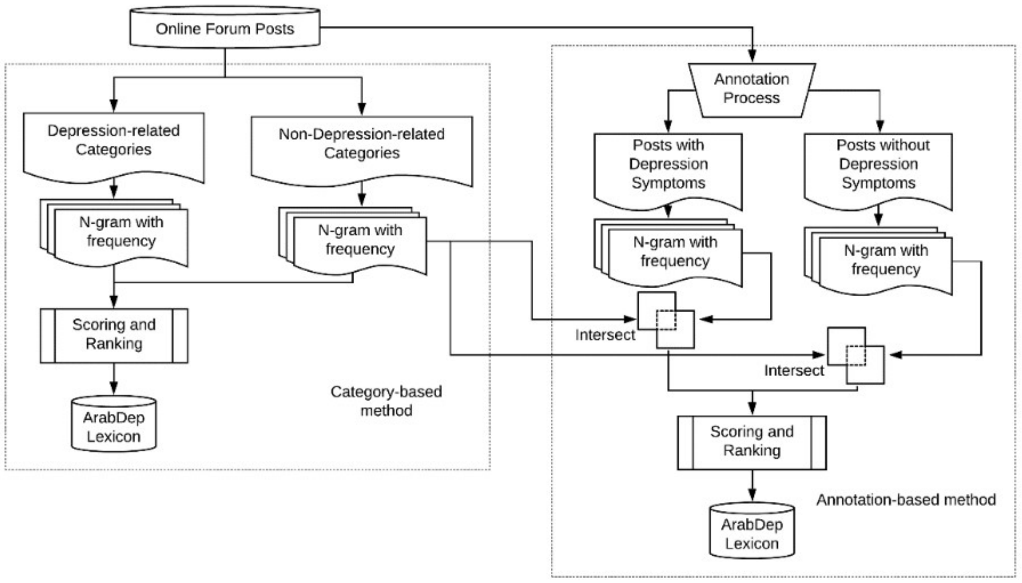
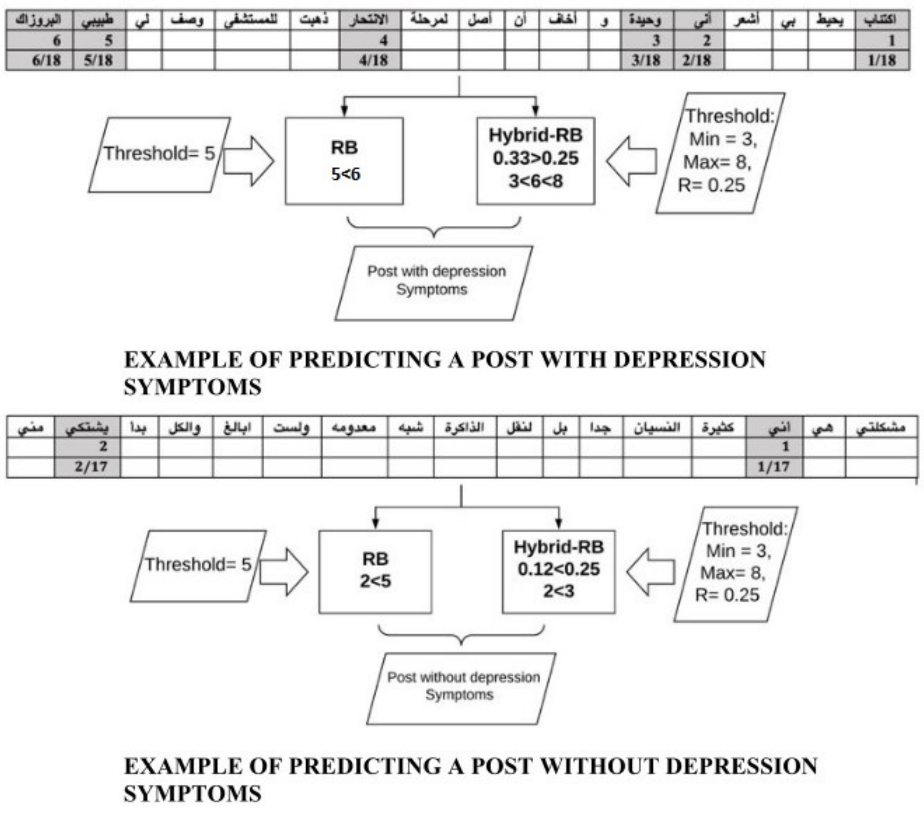


Figure 2.3: ArabDep creation methods

As shown in figure 2.3[26], The researchers used two methods for creating the Arab-Dep Lexicon category-based lexicon method where they apply the N-gram a model used in NLP [29] for both depression and non-depression related categories, they build three n-gram collection (unigram, bigram, trigram). The second method is using the psychologist class annotation mentioned above [26] to give high degree of granularity, the annotation where “clear”, “low depression”, “strong depression”, “no depression” and “could not decide”, the three annotation where labelled with ‘1’ and the last two with ‘0’, then repeat the process of the first method by using N-gram model.

2. **Rule-based algorithm:** Given a text T, the algorithm utilizes the lexicon to predict whether the sentence corresponds to depression symptoms or not based on a threshold. In figure 2.4, the first example contains words from the ArabDep lexicon, which are highlighted in grey, the number of depression-related words exceeds the threshold for both RB and Hybrid-RB function so the post corresponds to depression symptoms, the oposite in the second example. (See figure 2.4[26]).



البروزاك	طبيبي	لي	وصف	للمستشفى	ذهبت	الانتحار	لمرحلة	أصل	أن	أخاف	و	وحيدة	أني	أشعر	بي	يحيط	اكتئاب
6	5					4						3	2				1
6/18	5/18					4/18						3/18	2/18				1/18

EXAMPLE OF PREDICTING A POST WITH DEPRESSION SYMPTOMS

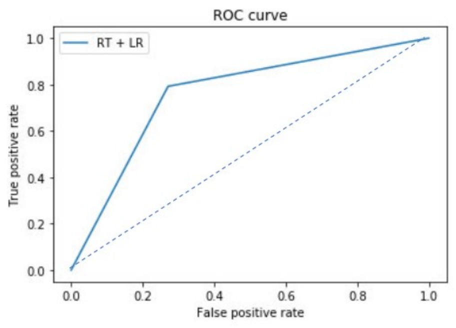
مني	يشتكي	بدا	والكل	ابالغ	ولست	معدومه	شبه	الذاكرة	لنقل	بل	جدا	النسيان	كثيرة	اني	هي	مشكلتي
	2													1		
	2/17													1/17		

EXAMPLE OF PREDICTING A POST WITHOUT DEPRESSION SYMPTOMS

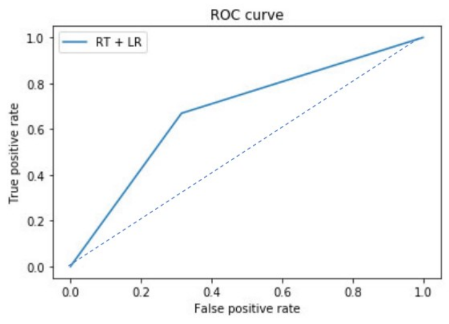
Figure 2.4: Rule-Based Algorithm

3. **Machine-learning-based approach:** The proposed framework is developed by using ADA boost[30], decision trees, k-nearest neighbour, random forest, support vector machine, and stochastic gradient descent.

Area under the curve-receiver operating characteristic curve for the proposed method in figure 2.5 [26].



(a)



(b)

Figure 2.5: (a): AUC = 0.78 for lexicon based approach (b): AUC= 72% for ML based approach

2.5.2 Predicting depression of social media user on different observation windows

In this study the researchers propose to predict user's depression via Sina Weibo data[30], Sina Weibo is one of the most popular Microblogging service providers in China[31]. They build classification models for differentiating participants with high and low scores on self-report depression measurement (CES-D)[32], and train regression models for predicting the CES-D score of any individual user.

After applying Breadth-first search algorithm to get data of the users selected ,The researchers instructed them to complete an online depression questionnaire CES-D composed of twenty items between August 1 and August 15 2013. The score of CES-D is presented in figure 2.6 [32].

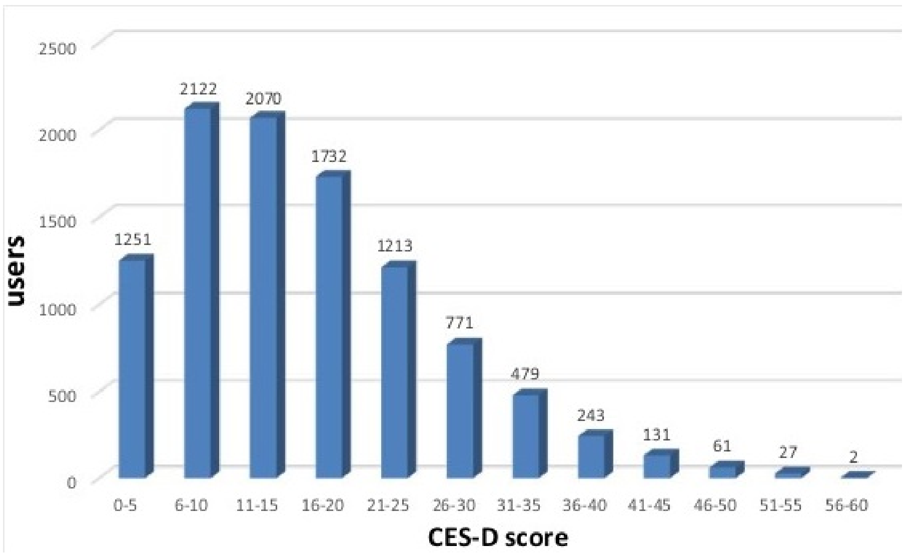


Figure 2.6: CES-D score

How depression CES-D is rated: The questionnaire Scores range from 0 (lowest) to 60 (highest), and patients are categorized into one of the following four groups: a) not depressed (0–9 points), b) mildly depressed (10–15 points), c) moderately depressed (16–24 points), or d) severely depressed (more than 25 points).

In this study the researchers used a Chinese text analysis software name's "WenXin" to process the text of Weibo posts user's[33], and they made matrices with the CES-D scores and the extracted features after they used calculated correlation between features and scores on CES-D to select sensitive features for indicating depression.

Models training and evaluation: They built the model using Logistic regression method and train the model with the calculation of the mean value and standard deviation of depression scores in each matrix. Meanwhile they built regression models using linear regression method. They run 10-fold cross-validation while training and testing classification and regression models. For the two models the researchers got the correlation coefficient between predicted scores and questionnaire scores ranged from 0.25 to 0.39 and the accuracy ranged from 63% to 82%.

2.6 Conclusion

In this chapter, we have defined the main foundations, methods and techniques of classification of feelings. The machine learning-based approach gives excellent results, this makes this approach adapt to the words used in the corpus. However, the major advantage is the need for manual annotation, which is difficult to achieve in large corpora. In the next chapter, we describe, step by step our adopted solution in order to improve and get better accuracy.

Chapter 3

Data preparation and global methodology

3.1 Introduction

Social networks have been developed as a great point for its users to communicate with their interested friends and share their opinions, photos, and videos reflecting their moods and feelings. This creates an opportunity to analyse social network data for user's feelings to investigate their moods and attitudes when they are communicating via these online tools. It gives opportunities for people working in the health sector to get insight of what might be happening at mental state of someone who posted a topic in a specific manner.

This chapter aims to detect whether the user is depressed, from the nature of his or her tweets and activity in the network. It can be further used to identify other mental illnesses and might even form an underlying infrastructure for new mechanisms that would help detect and limit depression diffusion in social networks. We have used sentiment 140 dataset [37] since there are no public depression dataset, we used a dataset scraped by twint [38]. The tweets are a text data, words in a sequence so we used Recurrent Neural networks to build a model that doesn't only consider the individual words, but the order they appear in.

3.2 Dataset

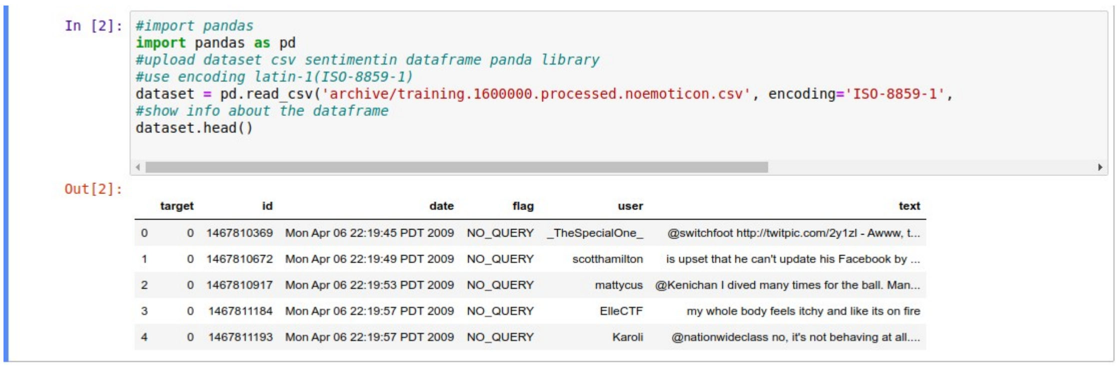
In this section, we present all the dataset used in this report.

3.2.1 Dataset 140 sentiment

Sentiment140 started as a class project from Stanford University. They assumed that any tweet with positive emoticons were positive, and tweets with negative emoticons were negative. They used the Twitter Search API to collect these tweets by using keyword search. We built this using the following technologies[39]:

- Twitter API,
- Amazon EC2 (for the backend),
- Google Closure (for the JavaScript library),
- Google Visualization API (for the annotated timeline),
- Google Charts API (for the pie and bar charts),
- Google Sites (for this documentation),
- Google Spreadsheets (for our feedback form),
- Google Analytics.

This dataset is a csv file with emoticons removed, that contains 1600000 tweets.



```
In [2]: #import pandas
        import pandas as pd
        #upload dataset csv sentimentin dataframe panda library
        #use encoding latin-1(ISO-8859-1)
        dataset = pd.read_csv('archive/training.1600000.processed.noemoticon.csv', encoding='ISO-8859-1',
        #show info about the dataframe
        dataset.head()
```

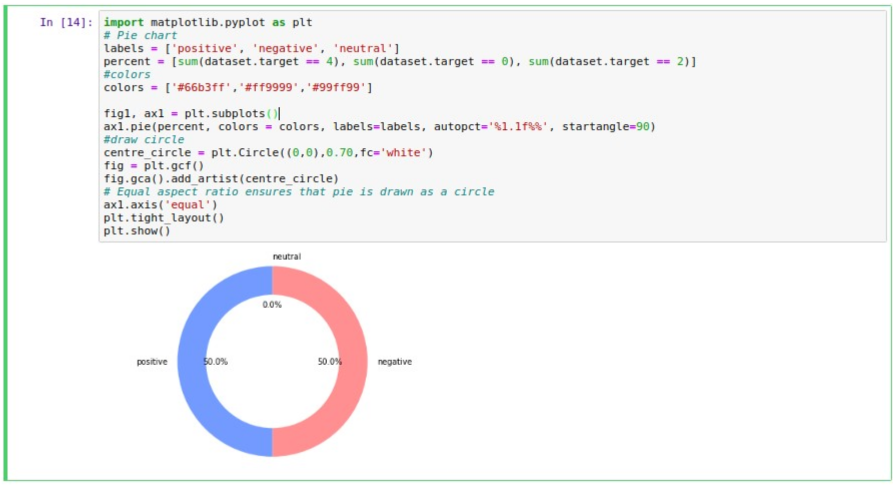
Out[2]:

	target	id	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Figure 3.1: Loading the dataset to panda dataframe

As shown in figure 3.1, the dataset has target attribute that defines the polarity of the tweet (0=negative, 2=neutral, 4=positive), the id and the date of the tweet and the two latest attributes are user and body of the tweet.

The percentage of the polarity in 140 sentiment dataset is showed in figure 3.2.

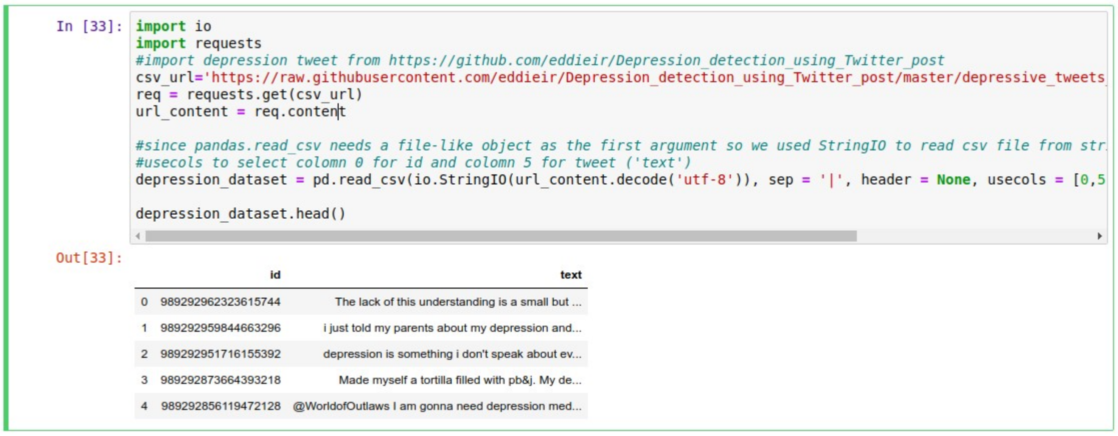


```
In [14]: import matplotlib.pyplot as plt
         # Pie chart
         labels = ['positive', 'negative', 'neutral']
         percent = [sum(dataset.target == 4), sum(dataset.target == 0), sum(dataset.target == 2)]
         #colors
         colors = ['#66b3ff','#ff9999','#99ff99']

         fig1, ax1 = plt.subplots()
         ax1.pie(percent, colors = colors, labels=labels, autopct='%1.1f%%', startangle=90)
         #draw circle
         centre_circle = plt.Circle((0,0),0.70,fc='white')
         fig = plt.gcf()
         fig.gca().add_artist(centre_circle)
         # Equal aspect ratio ensures that pie is drawn as a circle
         ax1.axis('equal')
         plt.tight_layout()
         plt.show()
```

Figure 3.2: The percentage of positive, negative and neutral tweet in 140 sentiment

3.2.2 Dataset depression tweets

Due to the lack of depressive tweets dataset, we used an existing one that is retrieved using the Twitter scraping tool TWINT [40]. The scraped tweets may contain tweets that do not indicate the user having depression because the dataset is just scraped with keyword ‘depression’ so we manually checked it for better testing results. This dataset is a csv file that contains 2345 tweets with two attributes, id and body of the tweet as showed in figure 3.3.



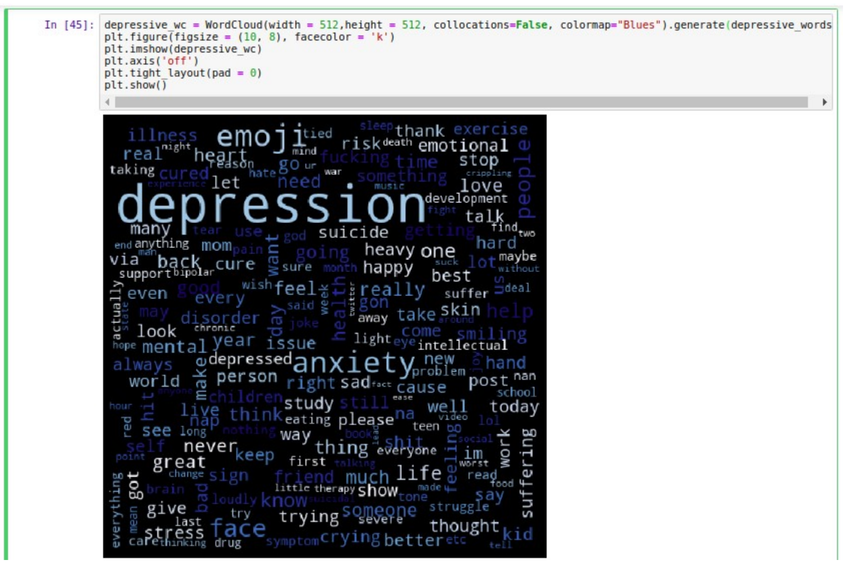
```
In [33]: import io
         import requests
         #import depression tweet from https://github.com/eddieir/Depression_detection_using_Twitter_post
         csv_url='https://raw.githubusercontent.com/eddieir/Depression_detection_using_Twitter_post/master/depressive_tweets
         req = requests.get(csv_url)
         url_content = req.content

         #since pandas.read_csv needs a file-like object as the first argument so we used StringIO to read csv file from str
         #usecols to select colomn 0 for id and colomn 5 for tweet ('text')
         depression_dataset = pd.read_csv(io.StringIO(url_content.decode('utf-8')), sep = '|', header = None, usecols = [0,5

         depression_dataset.head()
```

Out[33]:

	id	text
0	989292962323615744	The lack of this understanding is a small but ...
1	989292959844663296	i just told my parents about my depression and...
2	989292951716155392	depression is something i don't speak about ev...
3	989292873664393218	Made myself a tortilla filled with pb&j. My de...
4	989292856119472128	@WorldofOutlaws I am gonna need depression med...

Figure 3.3: Download an existing csv depression dataset

WordCloud the visual representation of text data in which the size of the words represents their frequency, in the figure 3.4, we spotted words that are indicative of depression in these tweets after the pre-treatment such as depression, anxiety, suffering, struggle, sad, suffer...



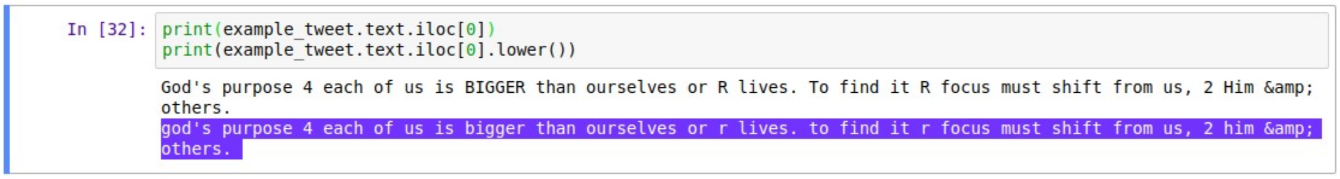
```
In [45]: depressive_wc = WordCloud(width = 512,height = 512, collocations=False, colormap="Blues").generate(depressive_words
         plt.figure(figsize = (10, 8), facecolor = 'k')
         plt.imshow(depressive_wc)
         plt.axis('off')
         plt.tight_layout(pad = 0)
         plt.show()
```

Figure 3.4: Wordcloud plotting

3.3 Data pre-processing

The pre-processing is the most important step in any machine learning algorithm, it transforms text into the form where the machine learning algorithm can perform. We mention all the functions in these steps :

- **Lowercasing:** We lowercase all the text data so as not to have different output for different variation for an input capitalization.

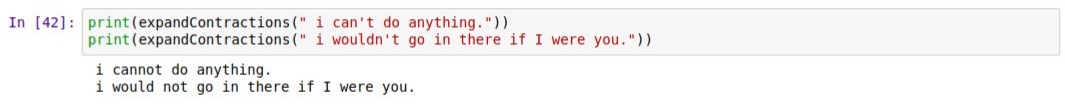


```
In [32]: print(example_tweet.text.iloc[0])
         print(example_tweet.text.iloc[0].lower())

         God's purpose 4 each of us is BIGGER than ourselves or R lives. To find it R focus must shift from us, 2 Him &amp;
         others.
         god's purpose 4 each of us is bigger than ourselves or r lives. to find it r focus must shift from us, 2 him &amp;
         others.
```

Figure 3.5: Example of lowercasing a random tweet

- **Convert accented characters:** Standardize and convert word with accent to words without accent mark.

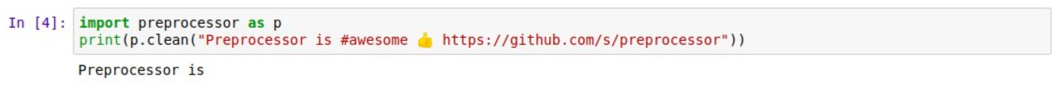
- **Expand contractions:** Expand contraction and short words for the purpose to standardize the text.



```
In [42]: print(expandContractions(" i can't do anything."))
         print(expandContractions(" i wouldn't go in there if I were you."))

          i cannot do anything.
          i would not go in there if I were you.
```

Figure 3.6: Example of applying expand contractions function

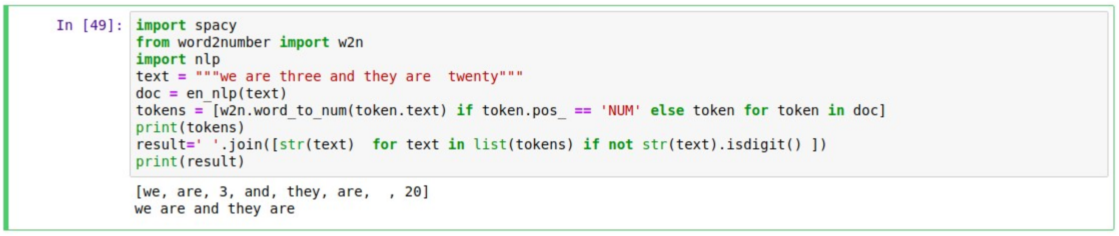
- **Pre-processor clean python library:** This library is specially designed for twitter data and has a basic function such as removing URL's, hashtags and remove mentions because all we have mentioned doesn't have any value for our adopted model.



```
In [4]: import preprocessor as p
        print(p.clean("Preprocessor is #awesome 👍 https://github.com/s/preprocessor"))

        Preprocessor is
```

Figure 3.7: Pre-processor clean python library

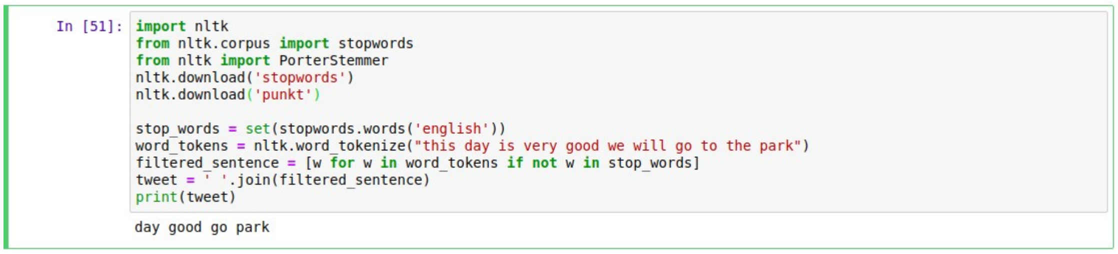
- **Remove numbers:** Since numbers do not contain any information about sentiment and depression, we remove them before processing the data, to do that, we first convert number words to digital form and then remove them.



```
In [49]: import spacy
         from word2number import w2n
         import nlp
         text = """we are three and they are  twenty"""
         doc = en_nlp(text)
         tokens = [w2n.word_to_num(token.text) if token.pos_ == 'NUM' else token for token in doc]
         print(tokens)
         result=' '.join([str(text)  for text in list(tokens) if not str(text).isdigit() ])
         print(result)

         [we, are, 3, and, they, are,  , 20]
         we are and they are
```

Figure 3.8: Example of removing numbers from text

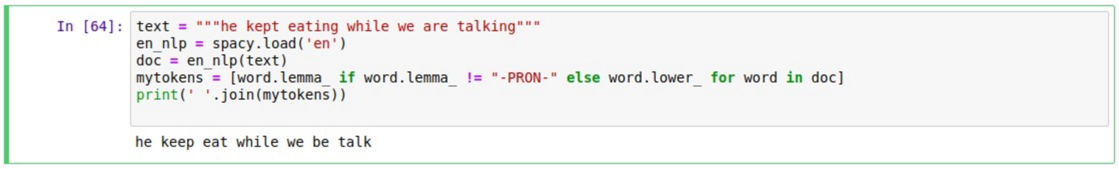
- **Remove stop words:** Stop words are the common words used in a language, and to get the best accuracy those words like 'we', 'I', 'they' are not helping for our adopted model, so we remove them from all our data. The figure 3.9 shows an example of a simple tweet processed with this function.



```
In [51]: import nltk
         from nltk.corpus import stopwords
         from nltk import PorterStemmer
         nltk.download('stopwords')
         nltk.download('punkt')

         stop_words = set(stopwords.words('english'))
         word_tokens = nltk.word_tokenize("this day is very good we will go to the park")
         filtered_sentence = [w for w in word_tokens if not w in stop_words]
         tweet = ' '.join(filtered_sentence)
         print(tweet)

         day good go park
```

Figure 3.9: Example of removing stop words method

- **Lemmatization:** Lemmatization is the method to convert the words to its base form, it is the process to reduce a given word to its root.



```
In [64]: text = """he kept eating while we are talking"""
         en_nlp = spacy.load('en')
         doc = en_nlp(text)
         mytokens = [word.lemma_ if word.lemma_ != "-PRON-" else word.lower_ for word in doc]
         print(' '.join(mytokens))

         he keep eat while we be talk
```

Figure 3.10: Example of applying lemmatization

3.4 Word embedding

Word embedding is a technique for representing a word context by a vector where words with the same meaning have a similar representation. There are many techniques to implement this approach, for our adopted model we used Word2Vec. We discuss about this pretrained google model in the following:

3.4.1 Word2Vec

This technique was proposed from Mikolov at Google in 2013, its purpose is to make the neural network based of the embedding more efficient [41]. Word2vec can be obtained using two methods, both of those techniques learn weights which act as word representations, we define them as follows:

Common Bag Of Words

This method tends to divine the probability of a word given a context as an input and tries to predict the word corresponding to the context. The figure 3.11 shows the architecture of common bag of words model where input is one hot encoded vector size V and the hidden layer contains N neurons and the output is a V length vector with elements being the SoftMax values.

- Wvn is the weight matrix that maps the input x to the hidden layer.
- W'nv is the weight matrix that maps the hidden layer outputs to the final output layer.

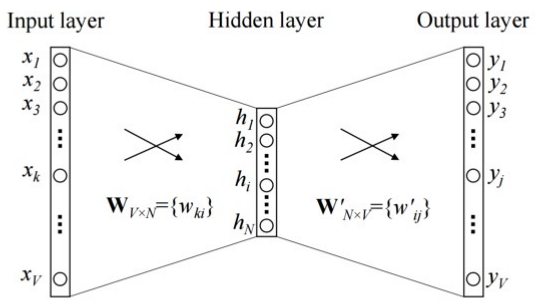
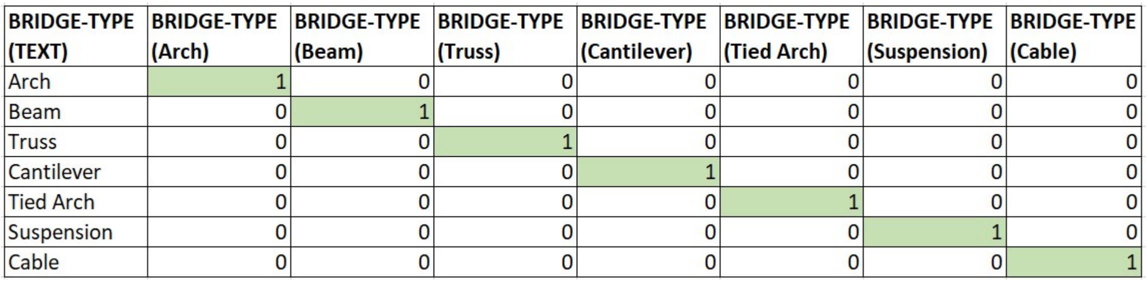


Figure 3.11: Common Bag Of Words schema

Skip Gram Model

This model follows the same topology as the first model (CBOW), it tries to predict the source context words given a target word. We will talk more about this topology because it is the one that is used to build the pre-trained model that we are going to use. First thing is to create a vocabulary of all the words in the text and then encode each word as the vector of the same dimension as the vocabulary where 1 represents the corresponding word in vocabulary and 0 for the rest. This technique is called one-hot encoding, a simple example is shown in the figure 3.12.



BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (Arch)	BRIDGE-TYPE (Beam)	BRIDGE-TYPE (Truss)	BRIDGE-TYPE (Cantilever)	BRIDGE-TYPE (Tied Arch)	BRIDGE-TYPE (Suspension)	BRIDGE-TYPE (Cable)
Arch	1	0	0	0	0	0	0
Beam	0	1	0	0	0	0	0
Truss	0	0	1	0	0	0	0
Cantilever	0	0	0	1	0	0	0
Tied Arch	0	0	0	0	1	0	0
Suspension	0	0	0	0	0	1	0
Cable	0	0	0	0	0	0	1

Figure 3.12: one-hot encoding vectors for an example

Input layer: The inputs of the network will be the vector representation of each word, for example if there are 10.000 unique words in the corpus, the one hot vector length is 10.000 where only one is '1' and the rest is 0.

Hidden layer: The context and the target pair are ready, the number of the neurons in the hidden layer will define how the model embed one hot vector inputs, for example if there are 300 dimension of the hidden layer, it means the model will create 300 features for each word in the vocabulary. As we suggested in the example above, the model have 10.000 inputs so the first hidden layer had 10.000 connection from each element of the inputs with only one input comes with '1' and the rest are zeros, so only one weight will be passed to the activation function for each word, and this is true for all 300 hidden layers, so each word will have its own set of 300 weights which will represent the vector of the word in the corpus.

Outputs layer: If we take the same example as above with 10.000 inputs, the outputs layer will have the same length as the input one where each element in the output vector has a value between 0 and 1 representing the probability of a particular word being nearby for the given input word. This layer has a SoftMax activation function to convert the output vector of the neuron network to a probability vector. The figure 3.13 shows a skip gram neural network architecture with an example for 'ants' as an input word.

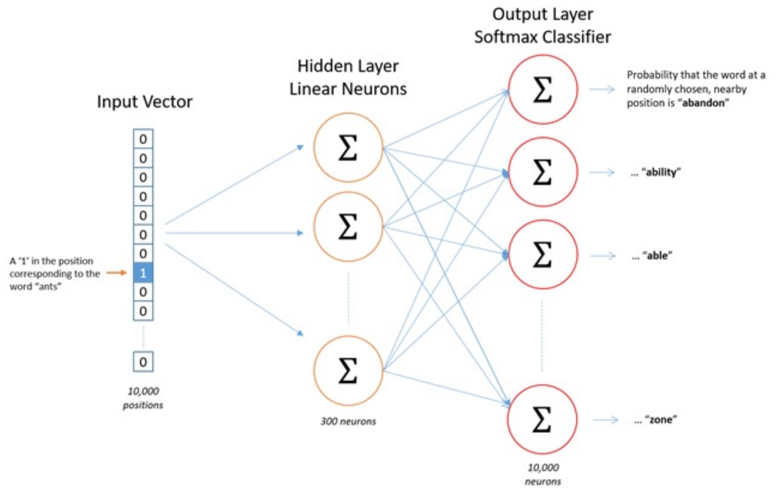


Figure 3.13: Neural network architecture for embedding words

3.4.2 Why not using our word2vec model

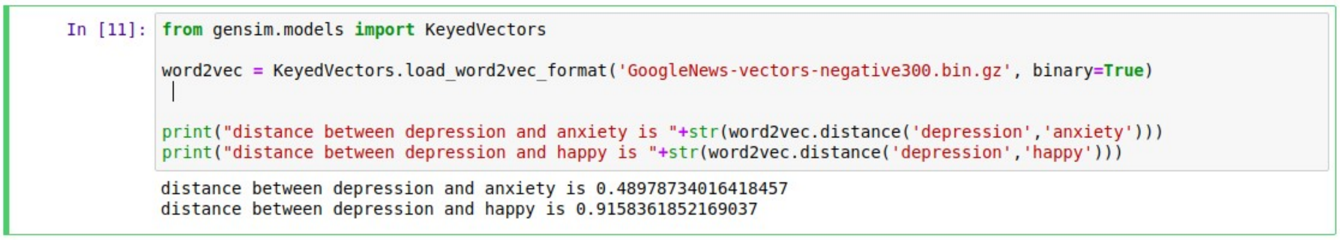
We mention two main reasons why not to use a scratch word embedding model:

Sparsity of training data: The major problem is to get a dataset that has a large volume of rare words, to build an efficient model we need to have a dataset that contains a rich vocabulary with frequent words.

Low resource: To build such a model of neural network we need a lot of time and well performed material to assure the efficiency of the results.

3.4.3 word2ev google news vectors

In our research, we used a pretrained word2vec model, it is word vector for a vocabulary of 3 million words with 300 features and phrases that trained on 100 billion words from google news dataset [41]. After loading the model with KeyedVectors module, we will perform some functions as follows.



```
In [11]: from gensim.models import KeyedVectors

         word2vec = KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin.gz', binary=True)
          |

         print("distance between depression and anxiety is "+str(word2vec.distance('depression','anxiety')))
         print("distance between depression and happy is "+str(word2vec.distance('depression','happy')))

         distance between depression and anxiety is 0.48978734016418457
         distance between depression and happy is 0.9158361852169037
```

Figure 3.14: Loading word2vec google pre-trained model with keyedVectors

In the figure 3.14, as we see after loading the pretrained model that if we try to get the cosine distance between the word depression and the word anxiety showed in equation (3.1) we get a 0.48, but if we try to get the cosine distance between the word depression and the word happy it returns 0.91. That means the word depression is very close to the word anxiety rather than the word happy.

We use cosine similarity for comparing the two vectors, which is defined as follows:

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \tag{3.1}$$

3.5 Global architecture of our implementation

In this section, we globally define the two models used in our adopted solution for detecting and predicting depression levels using twitter posts.

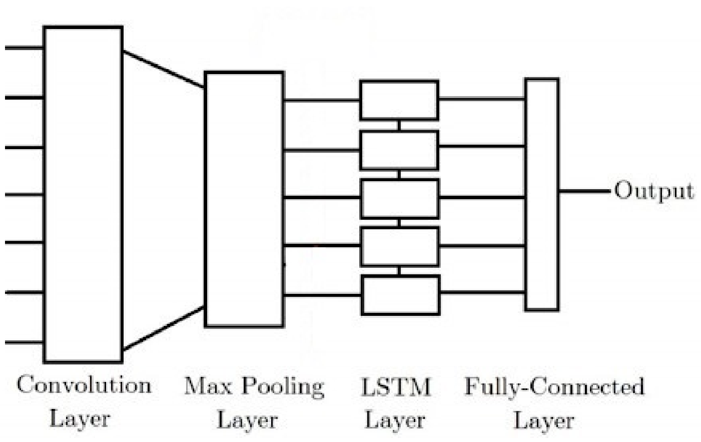


Figure 3.15: Global architecture of adopted methodology

As the figure 3.15 shows, we hybridized convolutional neural network and long short-term memory network in our implementation, we used CNN to extract high level features and LSTM algorithm which is used to classify the sentences based on the features already extracted by CNN.

3.5.1 Convolutional neural network

Is a type of neural network employed in image processing, however this model gives an excellent result at learning spatial structure from textual data. In the text processing instead of image pixels, the inputs of the CNN model are sentences or words represented as vectors, and each row of the matrix is word2vec word embedding vector that we have already created above, and when we talk about filter in text processing, it slides over rows of the matrix, for the sliding window it varies, in the example shown in the figure 3.16 an embedding matrix used with the shape of 7*5 and the size of the region is (4, 3, 2).

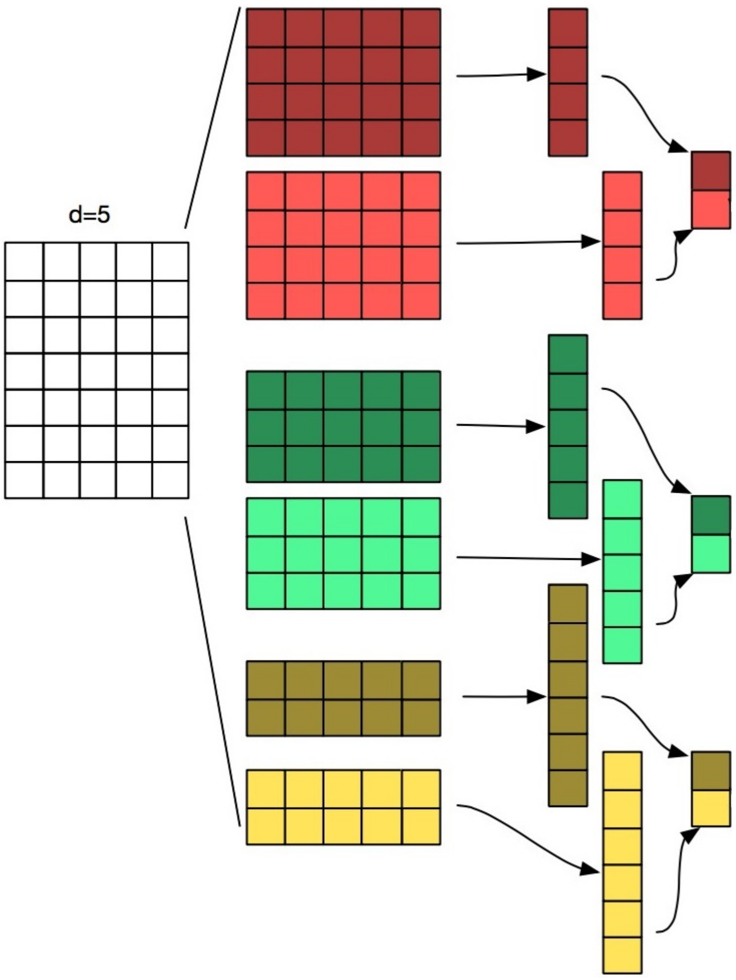


Figure 3.16: Architecture of convolutional neural network example

3.5.2 Long short-term memory model

It is an algorithm of recurrent neural network, it is capable to remember information for a long period of time. LSTM is divided into three modules [42]:

Forget gate

- Controls what information to throw away from memory.
- Decides how much from the past you should remember.

Input gate

- Controls what new information is added to the cell state from current input.
- Decides how much from this unit is added to the current state.

Output gate

- Conditionally decides what to output from the memory.
- Decides which part of the current cell makes it to the output.

Figure 3.17 presents a simple schema of long short-term memory model.

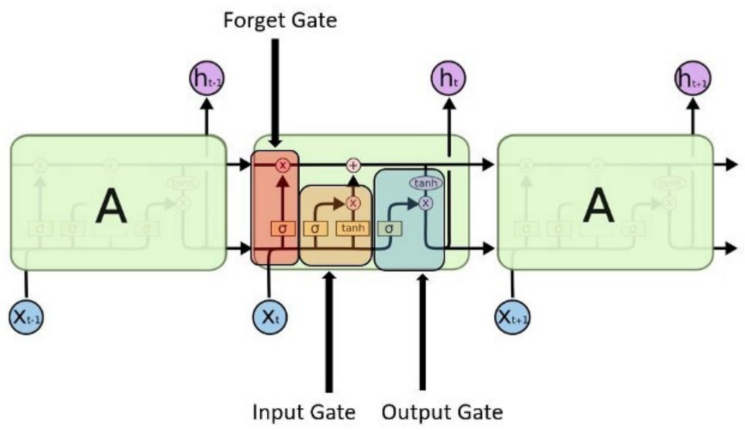


Figure 3.17: Long short-term memory model

3.6 Conclusion

In this chapter, we have presented a classification approach that contributes to the problem of depression prediction where we defined the dataset used in our work. Then we applied a pre-processing technique on these data which are an English language tweets, to minimize noise and remove ambiguity in order to improve the accuracy and quality of the classification. In the next chapter, we discuss in depth the method we propose to use and the results obtained when applying this method as well as a comparison with other similar models to choose the best classification method.

Chapter 4

Implementation and evaluation

4.1 Introduction

In this chapter, we present the tools and languages used to implement the hybrid convolutional neural network with long short-term memory classifier and its accuracy result.

We also compare our adopted model accuracy with other models like naive bayes classifier and tf-idf classifier.

4.2 Programming language

The programming language used in our project is python, according to TIOBE Community python is classed as the second in the top ten programming languages in the world [43]. It is used to develop GUI applications, websites, data science and data visualization and it is highly recommended in machine learning and artificial intelligence. We choose python due to its various libraries that allow to developers to perform complex tasks without the need to rewrite many lines in the code.

4.3 Frameworks

4.3.1 Anaconda

Anaconda is a distribution of packages built for data science, its environment manager is named conda which is used to install, uninstall and update packages. Conda is also used to create environment for isolating projects that use different versions. One of the most used packages in this distribution is jupyter notebook which is an online notebook that allows us to construct together computational information with narrative, multimedia and graphs[44].

4.4 Libraries

To create a predicting depression levels from social media posts project we have used built-in modules that helped us to get a standardized solution in our implementation, we will define some of them in the following:

4.4.1 Pandas

It is a software library written in python used especially in data science and data analysis. It offers operations for manipulating numerical tables and times series[45].

4.4.2 Numpy

Numpy is a python library written in C interpreted by Cpython, converted to bytecode and then executed[46], numpy allows programmers to process with arrays, it provides high performance in multi-dimensional array objects.

4.4.3 Keras

Keras is a high level neural networks API[47], written in Python and interfaceable with TensorFlow. It was developed with the aim of allowing rapid experiments. Being able to go from the idea to the result in the shortest time possible.

4.4.4 NLTK

The Natural Language Toolkit (NLTK) is a platform used to create text analysis programs[48].

4.4.5 Scikit-learn

Is a library developed in Python. It is dedicated to statistical learning and can be used as middleware, especially for prediction tasks. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy[49].

4.4.6 Gensim

It is an open source natural language processing library[50], used in various complex tasks such as building word vectors, topic identifications and document comparisons.

4.4.7 Matplotlib

It is a popular python package used for data visualisation[51]. it uses an API to embed plot in python applications.

4.5 Methodology

In this section, we demonstrate how we extracted the text features with convolutional neural network and get the most important features and feed them into the long short-term memory model to train the order of these features.

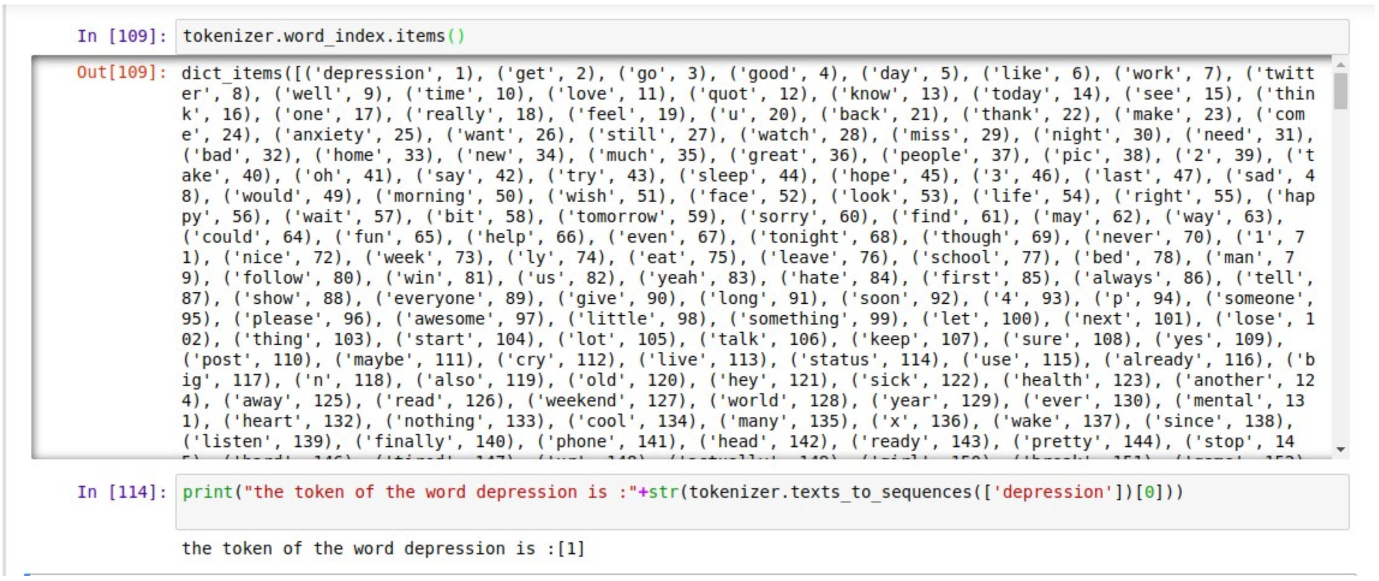
4.5.1 Convolutional neural network

Convolutional neural network is a popular algorithm used in computer vision, but the model also gives an efficient result in sequence processing. We will demonstrate how we applied the concept of convolution and filter model in our project. In the image processing field, CNN slides window function over a matrix, the image is represented as matrix where each entry corresponds to one pixel, and to get the full convolution it multiplies the value of filter mask with the matrix value and slides the filter over the whole image. On the other side in text processing instead of working with pixels, we have worked with word embedding vectors. The figure 4.1 shows a sample matrix of word encoding.

$$
\begin{array}{l}
\text{hello} \\
\text{there} \\
\text{texas} \\
\text{world} \\
\ldots
\end{array}
\left[
\begin{array}{ccccccc}
12 & 45 & 43 & 26 & 78 & 532 & \ldots \\
43 & 25 & 778 & 43 & 53 & 78 & \ldots \\
34 & 56 & 23 & 12 & 56 & 74 & \ldots \\
342 & 54 & 23 & 5 & 7 & 423 & \ldots \\
\ldots & & & & & &
\end{array}
\right]
$$

Figure 4.1: Example of word embedding matrix of a sentence

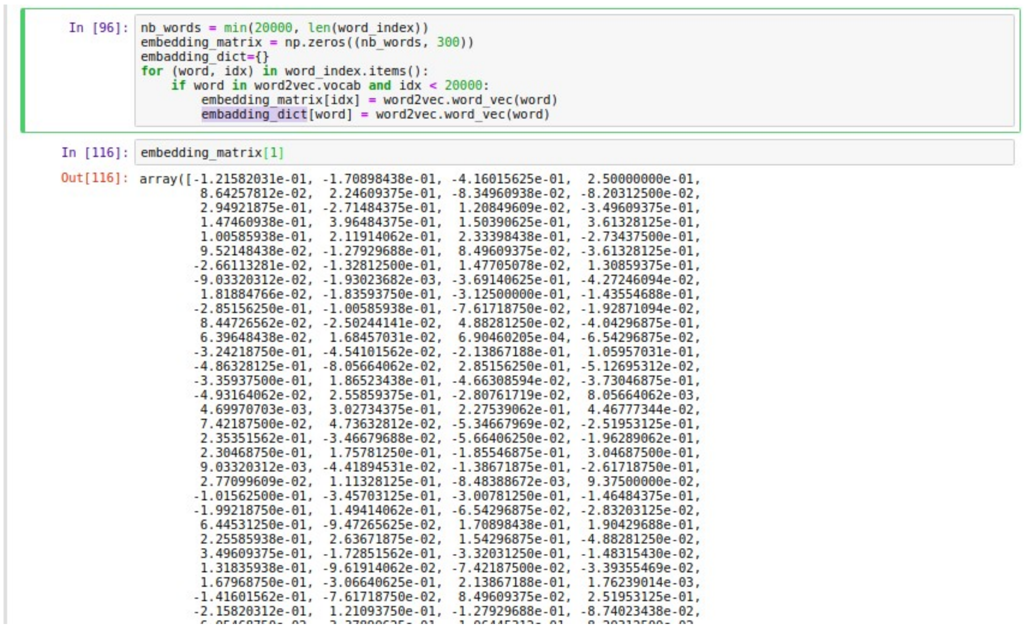
To get the matrix of the sentences inputs, we encoded each word as a unique token in the vectorization process that we talked about in the 3rd chapter, those tokens are assigned based on the frequency of the occurrence of a word in the dataset, for example if we take the word "depression" it will have the associated token 1 because it is the most common word in the dataset. We used the module Tokenizer from keras. preprocessing. text package to map every word in our vocabulary with the token after pre-processing and cleaning the dataset. In the figure 4.2 we show the tokens map and an example where we get the token of the word "depression".



```
In [109]: tokenizer.word_index.items()

Out[109]: dict_items([('depression', 1), ('get', 2), ('go', 3), ('good', 4), ('day', 5), ('like', 6), ('work', 7), ('twitt
          er', 8), ('well', 9), ('time', 10), ('love', 11), ('quot', 12), ('know', 13), ('today', 14), ('see', 15), ('thin
          k', 16), ('one', 17), ('really', 18), ('feel', 19), ('u', 20), ('back', 21), ('thank', 22), ('make', 23), ('com
          e', 24), ('anxiety', 25), ('want', 26), ('still', 27), ('watch', 28), ('miss', 29), ('night', 30), ('need', 31),
          ('bad', 32), ('home', 33), ('new', 34), ('much', 35), ('great', 36), ('people', 37), ('pic', 38), ('2', 39), ('t
          ake', 40), ('oh', 41), ('say', 42), ('try', 43), ('sleep', 44), ('hope', 45), ('3', 46), ('last', 47), ('sad', 4
          8), ('would', 49), ('morning', 50), ('wish', 51), ('face', 52), ('look', 53), ('life', 54), ('right', 55), ('hap
          py', 56), ('wait', 57), ('bit', 58), ('tomorrow', 59), ('sorry', 60), ('find', 61), ('may', 62), ('way', 63),
          ('could', 64), ('fun', 65), ('help', 66), ('even', 67), ('tonight', 68), ('though', 69), ('never', 70), ('1', 7
          1), ('nice', 72), ('week', 73), ('ly', 74), ('eat', 75), ('leave', 76), ('school', 77), ('bed', 78), ('man', 7
          9), ('follow', 80), ('win', 81), ('us', 82), ('yeah', 83), ('hate', 84), ('first', 85), ('always', 86), ('tell',
          87), ('show', 88), ('everyone', 89), ('give', 90), ('long', 91), ('soon', 92), ('4', 93), ('p', 94), ('someone',
          95), ('please', 96), ('awesome', 97), ('little', 98), ('something', 99), ('let', 100), ('next', 101), ('lose', 1
          02), ('thing', 103), ('start', 104), ('lot', 105), ('talk', 106), ('keep', 107), ('sure', 108), ('yes', 109),
          ('post', 110), ('maybe', 111), ('cry', 112), ('live', 113), ('status', 114), ('use', 115), ('already', 116), ('b
          ig', 117), ('n', 118), ('also', 119), ('old', 120), ('hey', 121), ('sick', 122), ('health', 123), ('another', 12
          4), ('away', 125), ('read', 126), ('weekend', 127), ('world', 128), ('year', 129), ('ever', 130), ('mental', 13
          1), ('heart', 132), ('nothing', 133), ('cool', 134), ('many', 135), ('x', 136), ('wake', 137), ('since', 138),
          ('listen', 139), ('finally', 140), ('phone', 141), ('head', 142), ('ready', 143), ('pretty', 144), ('stop', 14

In [114]: print("the token of the word depression is :"+str(tokenizer.texts_to_sequences(['depression'])[0]))

          the token of the word depression is :[1]
```

Figure 4.2: Print token vocabulary content

We encode the sentences with the value of the token of each word in the sentence. We define as default maximal length of every sentences as 140 words so we added 0 to all the sentences to make their length equal. For example if we take the word "depression" his token is 1 so if we try to get the first vector of the embedding matrix as the figure 4.3 shows we will get the vector representation of the word “depression”. We already discussed how to get the element of the vector in chapter 3.



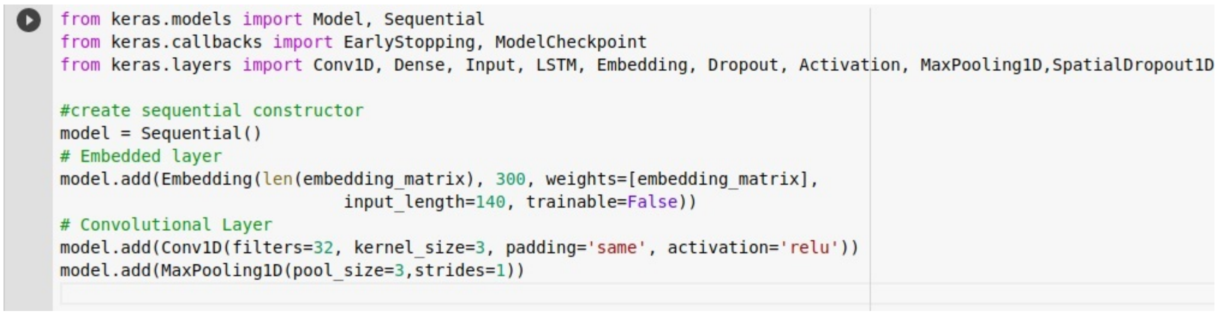
```
In [96]:  nb_words = min(20000, len(word_index))
          embedding_matrix = np.zeros((nb_words, 300))
          embadding_dict={}
          for (word, idx) in word_index.items():
              if word in word2vec.vocab and idx < 20000:
                  embedding_matrix[idx] = word2vec.word_vec(word)
                  embadding_dict[word] = word2vec.word_vec(word)

In [116]: embedding_matrix[1]

Out[116]: array([-1.21582031e-01, -1.70898438e-01, -4.16015625e-01,  2.50000000e-01,
                  8.64257812e-02,  2.24609375e-01, -8.34960938e-02, -8.20312500e-02,
                  2.94921875e-01, -2.71484375e-01,  1.20849609e-02, -3.49609375e-01,
                  1.47460938e-01,  3.96484375e-01,  1.50390625e-01,  3.61328125e-01,
                  1.00585938e-01,  2.11914062e-01,  2.33398438e-01, -2.73437500e-01,
                  9.52148438e-02, -1.27929688e-01,  8.49609375e-02, -3.61328125e-01,
                 -2.66113281e-02, -1.32812500e-01,  1.47705078e-02,  1.30859375e-01,
                 -9.03320312e-02, -1.93023682e-03, -3.69140625e-01, -4.27246094e-02,
                  1.81884766e-02, -1.83593750e-01, -3.12500000e-01, -1.43554688e-01,
                 -2.85156250e-01, -1.00585938e-01, -7.61718750e-02, -1.92871094e-02,
                  8.44726562e-02, -2.50244141e-02,  4.88281250e-02, -4.04296875e-01,
                  6.39648438e-02,  1.68457031e-02,  6.90460205e-04, -6.54296875e-02,
                 -3.24218750e-01, -4.54101562e-02, -2.13867188e-01,  1.05957031e-01,
                 -4.86328125e-01, -8.05664062e-02,  2.85156250e-01, -5.12695312e-02,
                 -3.35937500e-01,  1.86523438e-01, -4.66308594e-02, -3.73046875e-01,
                 -4.93164062e-02,  2.55859375e-01, -2.80761719e-02,  8.05664062e-03,
                  4.69970703e-03,  3.02734375e-01,  2.27539062e-01,  4.46777344e-02,
                  7.42187500e-02,  4.73632812e-02, -5.34667969e-02, -2.51953125e-01,
                  2.35351562e-01, -3.46679688e-02, -5.66406250e-02, -1.96289062e-01,
                  2.30468750e-01,  1.75781250e-01, -1.85546875e-01,  3.04687500e-01,
                  9.03320312e-03, -4.41894531e-02, -1.38671875e-01, -2.61718750e-01,
                  2.77099609e-02,  1.11328125e-01, -8.48388672e-03,  9.37500000e-02,
                 -1.01562500e-01, -3.45703125e-01, -3.00781250e-01, -1.46484375e-01,
                 -1.99218750e-01,  1.49414062e-01, -6.54296875e-02, -2.83203125e-02,
                  6.44531250e-01, -9.47265625e-02,  1.70898438e-01,  1.90429688e-01,
                  2.25585938e-01,  2.63671875e-02,  1.54296875e-01, -4.88281250e-02,
                  3.49609375e-01, -1.72851562e-01, -3.32031250e-01, -1.48315430e-02,
                  1.31835938e-01, -9.61914062e-02, -7.42187500e-02, -3.39355469e-02,
                  1.67968750e-01, -3.06640625e-01,  2.13867188e-01,  1.76239014e-03,
                 -1.41601562e-01, -7.61718750e-02,  8.49609375e-02,  2.51953125e-01,
                 -2.15820312e-01,  1.21093750e-01, -1.27929688e-01, -8.74023438e-02,
```

Figure 4.3: Matrix embedding of our vocabulary

The input of convolutional neural network will be the vector representation of each word in the sentences, for example if we take the sentence “the world is great “ the matrix input dimension will be 300*140 because for the input we will get only two words “world” and “great” and others will be removed because they are stop words and add 0 element to the vector to get all vectors equal.

Convolution over sentences

After we created the matrix input of each sentence we are able to apply slide window to this matrix. In our project we used 1D convolutional kernel with filter size equal to 3 so the matrix filter will be 3*300. Then the filter will move one by one down on the input sentences matrix and multiply each pair of word matrix with the weights value of the kernel and sum them to get one output value for each move. The final result of all moves will be a vector that contains 140 values. To extract more features in our network, we have used multiple kernels with maxpooling operation which is an operation to force the network to save only the maximum value of the output vector. The figure 4.4 below shows how we implemented this model using keras library.



```
from keras.models import Model, Sequential
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers import Conv1D, Dense, Input, LSTM, Embedding, Dropout, Activation, MaxPooling1D,SpatialDropout1D

#create sequential constructor
model = Sequential()
# Embedded layer
model.add(Embedding(len(embedding_matrix), 300, weights=[embedding_matrix],
                            input_length=140, trainable=False))
# Convolutional Layer
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=3,strides=1))
```

Figure 4.4: Convolutional neural network with keras

The paramaters used in this model are the following:

Sequential: from the documentation of keras create a linear stack of layers and pass the list of layer instance to the constructor[52].

Embedding layer: we used this layer to encode each word from the sentences input with their representation from wor2vec embedding matrix.

Cov1d: we used cov1d convolution neural network because the output of the previous layer will be a sequence of 1 dimension data. as parameter, we used 32 filters with 3 size of each filter, it means it will process 3 sentences in each filter.

MaxPooling1d: this layer used to select the maximum elements from the region of the feature map covered by the previous layer. Pool_size means it will look at two value in the map at a time, stride=1 means the pool will move by 1 value.

4.5.2 Long short-term memory

In the max pooling layer, we have kept the maximal values from each feature map. then those vectors will be fed to LSTM one by one. LSTM keeps contextual information on inputs by integrating a loop that allows information to flow from one step to the next, it keeps those information using a mechanism called cell state, these information are stored using neuron network called gates.

Forget gate

In this neuron the word from previous hidden state and the word in the current state will be passed through the sigmoid function and get a value between 0 and 1 and the word with a value closed to 1 will be forwarded (See figure 4.5).

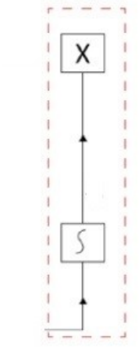


Figure 4.5: Forget gate

Input gate

As the concept of the previous gate, the words of the previous hidden state and the current state will be passed to function to squish values between -1 and 1 to help regulate the network, then the value will be multiplied with sigmoid output, the sigmoid function will decide which information is important to pass to the cell state (See figure 4.6).

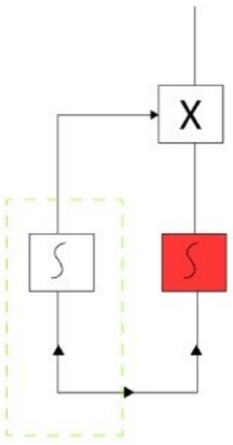


Figure 4.6: Input gate

Cell state

The forwarded information from the forget gate and input gate will be gathered. This neuron will act as a memory unit of the LSTM model (See figure 4.7).

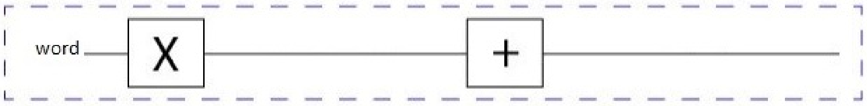


Figure 4.7: Cell state

Output gate

This state decides what the next hidden state will be (See figure 4.8).

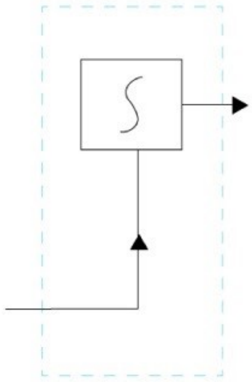
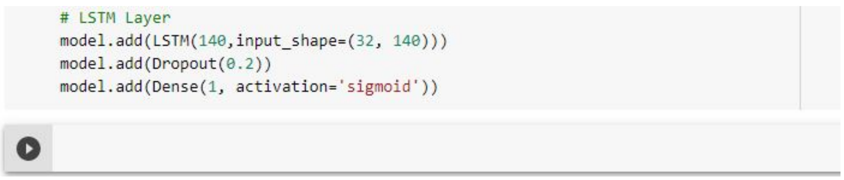


Figure 4.8: Output gate

For implementing this model, we have used LSTM method "from keras.Layer" module, for the input we feeded the output from the previous layer MaxPooling1D as Figure 4.8 shows.



```
# LSTM Layer
model.add(LSTM(140,input_shape=(32, 140)))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

Figure 4.9: LSTM layer implementation

The parameters of the model are the following :

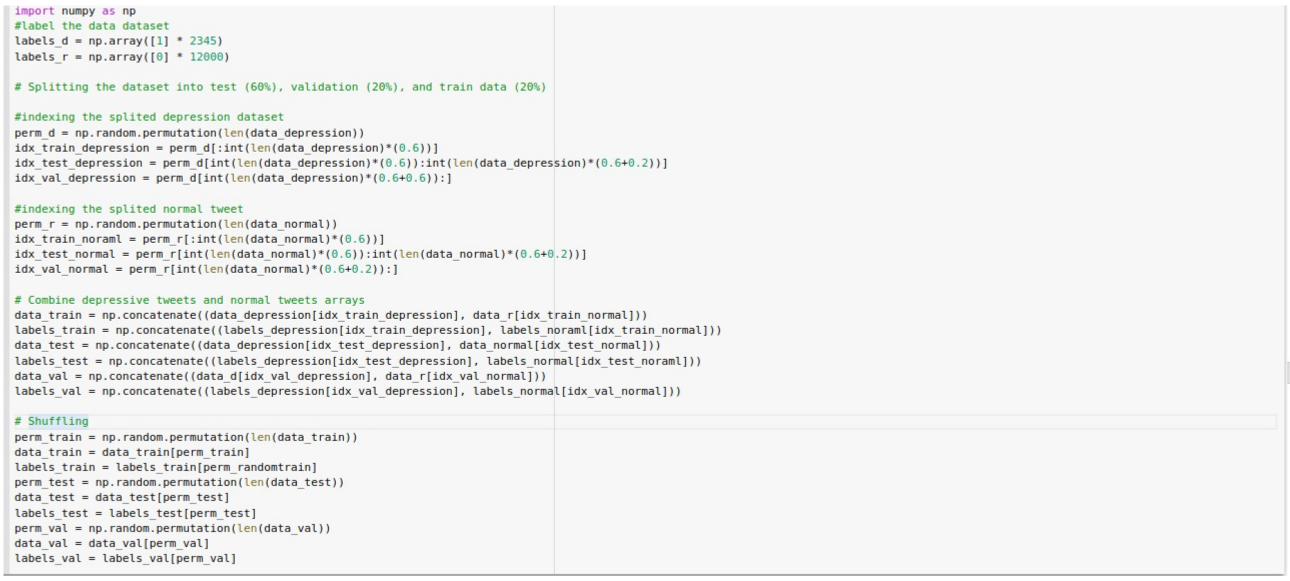
- **Input_dim** : this is dimensions of previous features. In our case 32*140 where 32 is the number of filters and 140 the size of each filter.
- **Dense layer** :It is Fully-connected layer where each neuron is connected to all of the neurons from the next layer. It implements the operation output = X * W + b where X is input to the layer in our case the output vector from LSTM model, and

W and b are weights and bias of the layer. W and b are actually the values that the neuron try to learn to get best accuracy.

4.6 Evaluation

In this section, we test the performance of our adopted model through which we quantify its prediction quality, and comparing the labelled data with the predicted data.

Before we apply training for our adopted model, we split our dataset into 3 categories: train, validation and test. As the figure 4.10 shows 60% is used for training, 20% for validation data and 20% for test data.



```
import numpy as np
#label the data dataset
labels_d = np.array([1] * 2345)
labels_r = np.array([0] * 12000)

# Splitting the dataset into test (60%), validation (20%), and train data (20%)

#indexing the splited depression dataset
perm_d = np.random.permutation(len(data_depression))
idx_train_depression = perm_d[:int(len(data_depression)*(0.6))]
idx_test_depression = perm_d[int(len(data_depression)*(0.6)):int(len(data_depression)*(0.6+0.2))]
idx_val_depression = perm_d[int(len(data_depression)*(0.6+0.6)):]

#indexing the splited normal tweet
perm_r = np.random.permutation(len(data_normal))
idx_train_noraml = perm_r[:int(len(data_normal)*(0.6))]
idx_test_normal = perm_r[int(len(data_normal)*(0.6)):int(len(data_normal)*(0.6+0.2))]
idx_val_normal = perm_r[int(len(data_normal)*(0.6+0.2)):]

# Combine depressive tweets and normal tweets arrays
data_train = np.concatenate((data_depression[idx_train_depression], data_r[idx_train_normal]))
labels_train = np.concatenate((labels_depression[idx_train_depression], labels_noraml[idx_train_normal]))
data_test = np.concatenate((data_depression[idx_test_depression], data_normal[idx_test_normal]))
labels_test = np.concatenate((labels_depression[idx_test_depression], labels_normal[idx_test_noraml]))
data_val = np.concatenate((data_d[idx_val_depression], data_r[idx_val_normal]))
labels_val = np.concatenate((labels_depression[idx_val_depression], labels_normal[idx_val_normal]))

# Shuffling
perm_train = np.random.permutation(len(data_train))
data_train = data_train[perm_train]
labels_train = labels_train[perm_randomtrain]
perm_test = np.random.permutation(len(data_test))
data_test = data_test[perm_test]
labels_test = labels_test[perm_test]
perm_val = np.random.permutation(len(data_val))
data_val = data_val[perm_val]
labels_val = labels_val[perm_val]
```

Figure 4.10: Splitting data into train, validation and test data

4.6.1 The model accuracy

Accuracy is a way to measure how the model classifies the data correctly, it is exactly the number of correctly predicted data points out of all the data points, mathematically it is the number of the true predicted value divided by the number of the all data.

```
plt.plot(hist.history['acc'])
plt.plot(hist.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')

plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```

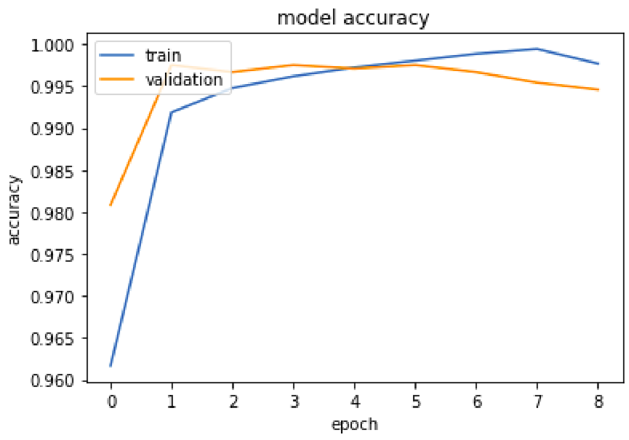
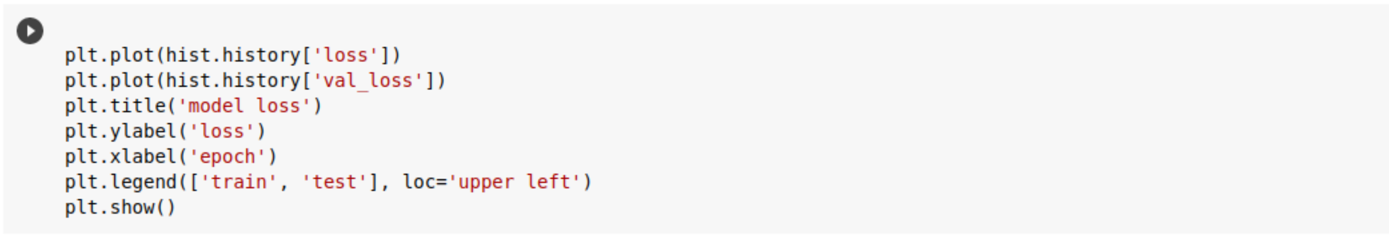


Figure 4.11: Plotting accuracy value over epoch training

Figure 4.11 shows that the accuracy score for both train and validation data are approximate, which means the model predicts the class of evaluation data as well as the trained data.

4.6.2 The model loss

The loss is calculated in the learning process, it is interpreted as how the model gets the right result over each epoch. It is the sum of errors for each epoch. In the figure 4.12, the average loss over each batch of training data and the complete model loss value of evaluation data over epoch are very small which means there is no overfitting in our training model. We used the parameter early stopping callback to avoid this problem, it stops the training when the value loss increases.



```
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

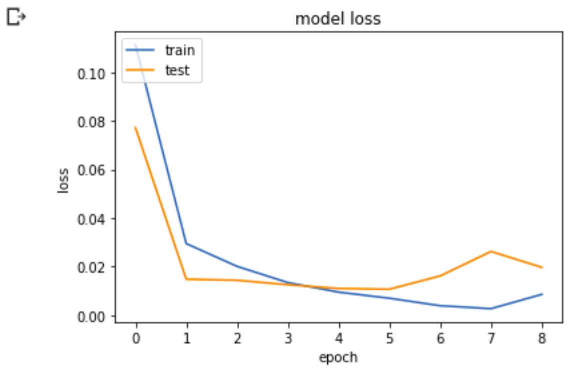
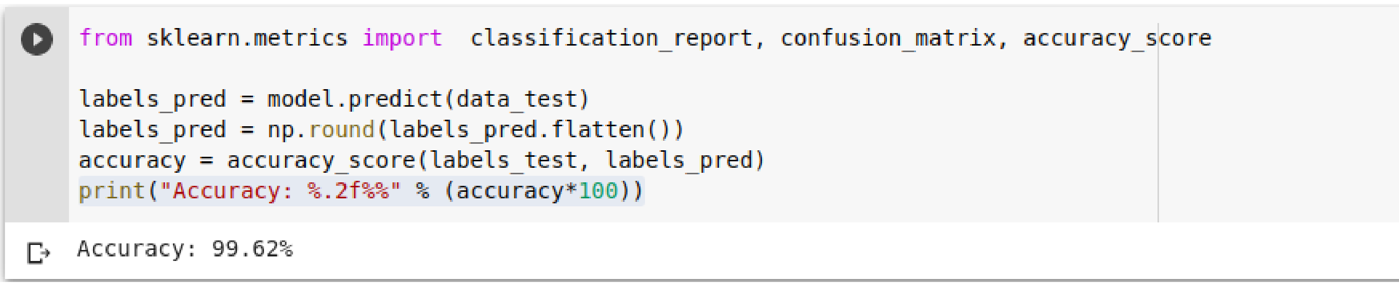


Figure 4.12: Plotting loss value over epoch training

4.6.3 Prediction accuracy of our adopted model

After the training phase has completed, we predict the test data class using the trained model then we get the accuracy score of the orediction class by the original labels which is 99.62 % (See figure 4.13).



```
from sklearn.metrics import  classification_report, confusion_matrix, accuracy_score

labels_pred = model.predict(data_test)
labels_pred = np.round(labels_pred.flatten())
accuracy = accuracy_score(labels_test, labels_pred)
print("Accuracy: %.2f%%" % (accuracy*100))
```

```
Accuracy: 99.62%
```

Figure 4.13: Accuracy value of predicting test data with our adopted model

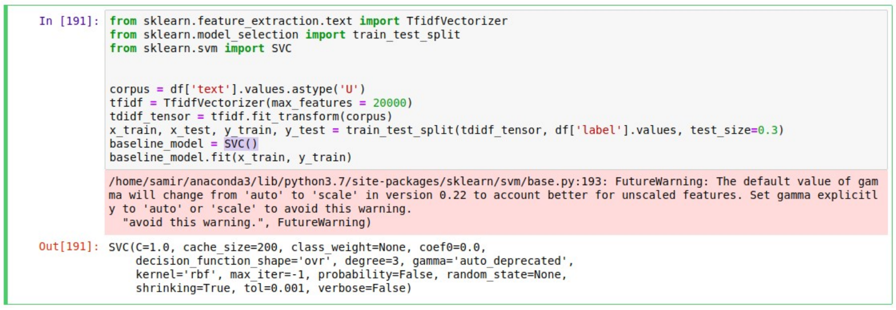
We can interpret this value as our trained model, it can detect depression from new twitter posts that we don't have in the dataset.

4.7 Baseline models

In this section we present the comparaison of our adopted model with some baseline models such as tf-idf and naive bayes classifiers.

4.7.1 Tf-idf

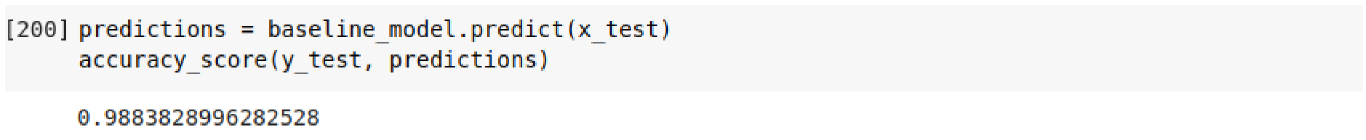
Term frequency-inverse dense frequency is a model that calculates how important the word is by weighing its frequency of occurence in the document and computing how often the same word occurs in other documents. If a word occurs many times in a particular document but not in others, then it might be highly relevant to that particular document and therefore is assigned more important[10]. In the figure 4.14, we implement the tf-idf model using sklearn library.



```
In [191]: from sklearn.feature_extraction.text import TfidfVectorizer
          from sklearn.model_selection import train_test_split
          from sklearn.svm import SVC


          corpus = df['text'].values.astype('U')
          tfidf = TfidfVectorizer(max_features = 20000)
          tdidf_tensor = tfidf.fit_transform(corpus)
          x_train, x_test, y_train, y_test = train_test_split(tdidf_tensor, df['label'].values, test_size=0.3)
          baseline_model = SVC()
          baseline_model.fit(x_train, y_train)

          /home/samir/anaconda3/lib/python3.7/site-packages/sklearn/svm/base.py:193: FutureWarning: The default value of gam
          ma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitl
          y to 'auto' or 'scale' to avoid this warning.
            "avoid this warning.", FutureWarning)

Out[191]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
              decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
              kernel='rbf', max_iter=-1, probability=False, random_state=None,
              shrinking=True, tol=0.001, verbose=False)
```

Figure 4.14: Implementation of Tf-idf classifier using sklearn library

After training the model we get 98% accuracy score in predicting the class label of the test data (See figure 4.15).

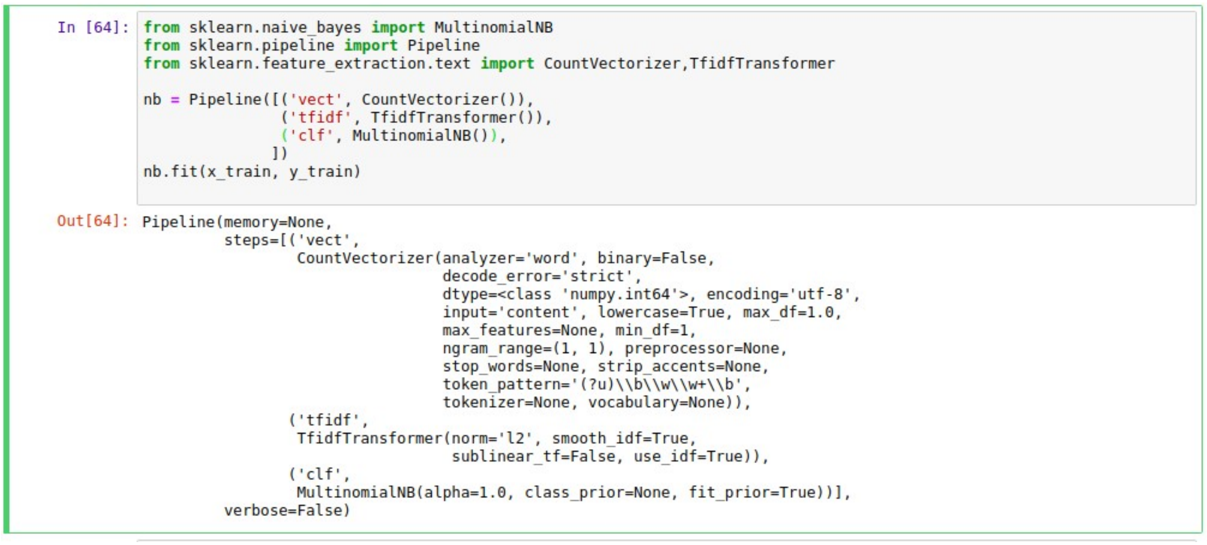


```
[200] predictions = baseline_model.predict(x_test)
      accuracy_score(y_test, predictions)

      0.9883828996282528
```

Figure 4.15: Accuracy value of predicting test data with our Tf-idf classifier

4.7.2 Naive bayes

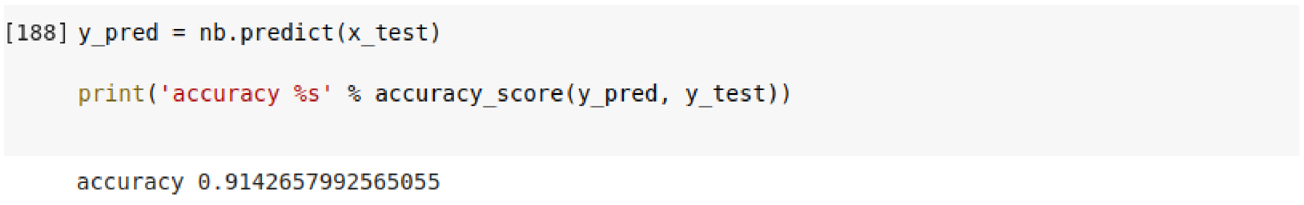
This model applies bayes theorem, this theorem shows the relationship between conditional probabilities and marginal probabilities in a probability distribution for a random variable[53]. In our case it describes the probability of the tweet based on sentence words. We have implemented the naive bayes classifier using MultinomialNB method from package sklearn.naive_bayes and we feed the model with the data that is already split. The figure 4.16 shows that we used sklearn.Pipline to encapsulate Countvectorizer, Tfidf-Transformer and MultinomialNBin one object, each transformer will be fitted with the output of the previous transformer.



```
In [64]: from sklearn.naive_bayes import MultinomialNB
         from sklearn.pipeline import Pipeline
         from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer

         nb = Pipeline([('vect', CountVectorizer()),
                        ('tfidf', TfidfTransformer()),
                        ('clf', MultinomialNB()),
                       ])
         nb.fit(x_train, y_train)

Out[64]: Pipeline(memory=None,
                  steps=[('vect',
                          CountVectorizer(analyzer='word', binary=False,
                                          decode_error='strict',
                                          dtype=<class 'numpy.int64'>, encoding='utf-8',
                                          input='content', lowercase=True, max_df=1.0,
                                          max_features=None, min_df=1,
                                          ngram_range=(1, 1), preprocessor=None,
                                          stop_words=None, strip_accents=None,
                                          token_pattern='(?u)\\b\\w\\w+\\b',
                                          tokenizer=None, vocabulary=None)),
                         ('tfidf',
                          TfidfTransformer(norm='l2', smooth_idf=True,
                                           sublinear_tf=False, use_idf=True)),
                         ('clf',
                          MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True))],
                  verbose=False)
```

Figure 4.16: Implementation of naive bayes model with sklearn library

After the training we get 91.4% for the accouracy of the naive bayes model, the figure below shows the output of the accuracy_score function with comparing the label output of the model and real class of the data.



```
[188] y_pred = nb.predict(x_test)

      print('accuracy %s' % accuracy_score(y_pred, y_test))

      accuracy 0.9142657992565055
```

Figure 4.17: Accuracy value of predicting test data with naive bayes classifier

4.8 Discussion

In this section we illustrate the different aspects that we used to compare our adopted model with the models mentioned above.

4.8.1 Confusion Matrix

Is a summary of the results of predictions in the classification problem, this matrix helps to understand how the classification model is confused when making predictions. This makes it possible to know which mistakes were made (see table 4.1).

	predicted depression	predicted no depression
depression	TP	FP
no depression	FN	TN

Table 4.1: Confusion matrix

Confusion matrix parameters are the following:

- True positive TP: positive class considered positive,
- True negative TN: negative class considered negative,
- False positive FP: negative class considered positive,
- False negative FN: positive class considered negative.

With the confusion matrix, we can calculate accuracy, precision, recall and f1-score which are defined as follows:

Accuracy: Is defined as the percentage of correct predictions for the test data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision: Is the ratio of correctly predicted positive class to the total predicted positive labels.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall: Is the ratio of correctly predicted positive labels to the all observations with the same class.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

F1-score: It is the harmonic average of the combination recall value and accuracy value.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

The different evaluation results are the following, (See table 4.2).

	CNN-LSTM		TF-IDF		Naive-Baise	
Accuracy	0.99		0.98		0.91	
Precesion	0.96	0.99	0.99	0.98	0.90	1.0
Recal	0.96	0.99	0.93	0.99	1.0	0.48
F1-Score	0.96	0.99	0.96	0.99	0.95	0.65
	depression	no depression	depression	no depression	depression	no depression

Table 4.2: Evaluation results

4.8.2 Time of prediction

Is the time that the model takes to predict the output of test data. The advantage of baseline models is that they are much faster then our adopted model, but they give less accuracy, the table 4.3 shows the different times that each model take to predict the class label of the test data.

Hybrid Model (our implementation)	6.67 seconds
NaiveBayse	0.051 seconds
TF-IDF	2.842 secondes

Table 4.3: Time Prediction Comparaison

In our implementation, we used google colab free version with the following specifications:

- **Processor:** Intel(R) Xeon(R) CPU 2.20GHz.
- **Radom Access memory:** 12GB.
- **Hard disk memory:** 108GB.
- **Graphics processing unit:**NVIDIA Tesla K80.

4.9 Conclusion

In this chapter, we have illustrated the different languages and development tools used to implement the hybrid convolutional neural network and long short-term memory model. We also presented the evaluation parameters applied to this model, with the objective of testing them in terms of operation and the viability of depression detection responses. We also implemented other baseline models like tf-idf and naive Bayes classifiers to compare them with the proposed model. The two other models are better in the time of prediction and the time of training but they give less accuracy and precision. The hybrid convolutional neural network and long short-term memory model is better in terms of accuracy and precision on the divided dataset.

General conclusion

In the midst of the global COVID-19 pandemic. The stress of social isolation, the worry about jobs, money, and health, and the profound feelings of loss that many of us are experiencing at the moment can trigger depression.

During this crisis, social media becomes the most used space for public to express their feelings and moods. Our main goal is to use our acknowledgement in artificial intelligence and machine learning to develop a model that can detect and predict depression levels from social media posts.

In order to achieve our goal, we have tried to understand the concepts of the convolutional neural network and long short-term memory to hybridize those methods in one model.

In this study, we presented how the famous convolutional neural network algorithm use filters to extract features from a matrix and long short-term memory which is a particular type of recurrent neural network that can use those features to classify and predict the input sentences. We also presented all the necessary steps in the data preprocessing and words embedding. Then we used those encoded data to implement the hybrid system in order to train the model to detect and predict depression from social media posts.

The adopted model that we have implemented using the python language is the result of a both theoretical and practical work. It gives a better performance in prediction accuracy comparing to other baseline models like tf-idf and naive Bayes that we implemented in this study.

For future work, we can:

- apply this model in sentiment analysis,
- use immense depression dataset labelled by psychologists,
- use our adopted model in other languages,
- improve prediction time of our adopted model.

Bibliography

[1] C. Clayton Childress. All Media Are Social. sagepub. published 14/02/2012.

[2] Gary Perkins. Internet Bulletin Board Systems: How the BBS is still relevant today. goodreads. published in 26/07/2011.

[3] Tom Johnstone, Mohammed Al-Mosaiwi. In an Absolute State: Elevated Use of Absolutist Words Is a Marker Specific to Anxiety, Depression, and Suicidal Ideation. sagepub. 05/2018.

[4] Jared D. Bernard, Jenna L. Baddeley, Benjamin F. Rodriguez, Philip A. Burke. Depression, Language, and Affect: An Examination of the Influence of Baseline Depression and Affect Induction on Language. sagepub. 02/06/2015.

[5]

[6] Norah Saleh Alghamdi, Hanan A Hosni Mahmoud, Ajith Abraham, Samar Awadh Alanazi, Laura Garcia-Hernande. Predicting Depression Symptoms in an Arabic Psychological Forum. IEEE. published 18/03/2020.

[7] Mowery, Danielle and Bryan, Craig and Conway, Mike. Towards Developing an Annotation Scheme for Depressive Disorder Symptoms: A Preliminary Study using Twitter Data. researchgate. published 01/2015.

[8] L Baer , D G Jacobs, J Meszler-Reizes, M Blais, M Fava, R Kessler, K Magruder, J Murphy, B Kopans, P Cukor, L Leahy, J O'Laughlen. Development of a brief screening instrument: the HANDS. pubmed. published 2000.

[9] Lowe, Bernd and Kroenke, Kurt and Herzog, Wolfgang and Grafe, Kerstin. Measuring depression outcome with a brief self-report instrument: Sensitivity to change of the Patient Health Questionnaire (PHQ-9). researchgate. published 08/2004.

[10] Daniel Jurafsky, James H Martin. Speech and Language Processing. stanford.edu. 02/10/2019.

[11] Q Hu, A Li, F Heng, J Li and T Zhu, Predicting Depression of Social Media User on Different Observation Windows. IEEE. published 2015.

[12] Lenore Sawyer Radloff. The CES-D Scale: A Self-Report Depression Scale for Research in the General Population. sagepub. published 06/01/1997.

[13] Morgan Kaufmann. Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) 3rd Edition. sciencedirect. published 09/06/2011.

[14] https://www.skillsyouneed.com/ips/what-is-communication.html, consulted 02/12/2019.

[15] https://medium.com/open-collective/social-coop-a-cooperative-decentralized-social-network-c10980c9ed91, consulted 03/12/2019.

[16] https://www.cbsnews.com/news/what-is-web-20/, consulted 08/12/2019.

[17] https://blog.hootsuite.com/types-of-social-media/, consulted 10/12/2019.

[18] https://www.yelp-support.com/article/What-is-Yelp-Connect?l=en_US, consulted 10/12/2019.

[19] https://www.compuserve.com/home/about.jsp, consulted 12/12/2019.

[20] https://www.cnbc.com/2015/05/12/timeline-aol-through-the-years.html, consulted 12/12/2019.

[21] https://www.cbsnews.com/pictures/then-and-now-a-history-of-social-networking-sites/2/, consulted 15/12/2019.

[22] https://news.linkedin.com/about-us#statistics, consulted 15/12/2019, consulted 18/12/2019.

[23] https://www.wsj.com/articles/facebooks-timeline-15-years-in-11549276201, consulted 20/12/2019.

[24] https://www.omnicoreagency.com/twitter-statistics/, consulted 23/12/2019.

[25] https://wearesocial.com/blog/2018/01/global-digital-report-2018, consulted 25/12/2019.

[26] https://www.openthegovernment.org/wp-content/uploads/other-files/Messaging%20One-Pager.pdf, consulted 25/12/2019.

[27] https://www.cnbc.com/2019/11/13/facebook-removed-3point2-billion-fake-accounts-between-apr-and-sept.html?&qsearchterm=twitter%20fake%20account, consulted 27/12/2019.

[28] https://time.com/5057998/germany-china-cyberspying-linkedin/, consulted 28/12/2019.

[29] https://www.who.int/news-room/fact-sheets/detail/depression, consulted 29/12/2019.

[30] https://www.who.int/news-room/fact-sheets/detail/depression, consulted 29/12/2019.

https://www.aappublications.org/news/2017/05/04/PASAngst050417, consulted 03/04/2020.

[31] https://www.iflscience.com/health-and-medicine/people-with-depression-use-language-differently-heres-how-to-spot-it/, consulted 10/04/2020.

[32] https://www.nafsany.cc/, consulted 28/04/2020.

[33] https://www.weibo.com/, consulted 25/05/2020.

[34] https://weibo.com/, consulted 30/05/2020

[35] https://machinelearningmastery.com/linear-regression-for-machine-learning/, consulted 10/06/2020.

[36] https://www.mcgill.ca/newsroom/channels/news/medium-machine-learning-and-mental-health-300119, consulted 15/05/2020.

[37] http://help.sentiment140.com/for-students, consulted 21/05/2020.

[38] https://github.com/twintproject/twint, consulted 10/04/2020.

[39] Alec Go, Richa Bhayani, Lei Huang. stanford. published 2009.

[40] https://github.com/peijoy/DetectDepressionInTwitterPosts, consulted 20/05/2020.

[41] https://code.google.com/archive/p/word2vec/, consulted 10/06/2020.

[42] https://medium.com/ahmetozlu93/long-short-term-memory-lstm-networks-in-a-nutshell-363cd470cca, consulted 03/08/2020.

[43] https://www.tiobe.com/tiobe-index/, consulted 02/07/2020.

[44] https://docs.anaconda.com/anaconda/install/, consulted 02/05/2020.

[45] https://pandas.pydata.org/docs/, consulted 04/05/2020.

[46] https://numpy.org/doc, consulted 05/05/2020.

[47] https://faroit.com/keras-docs/1.2.0/, consulted 12/07/2020.

[48] https://www.nltk.org/, consulted 13/07/2020.

[49] https://devdocs.io/scikit_learn/, consulted 13/08/2020

[50] https://radimrehurek.com/gensim/auto_examples/index.html, consulted 26/08/2020.

[51] https://matplotlib.org/3.3.3/contents.html, consulted 14/07/2020.

[52] https://keras.io/guides/sequential_model/, consulted 10/10/2020.

[53] https://globalaihub.com/sentiment-analysis-with-naive-bayes/, consulted 03/10/2020.