AN EFFICIENT CLASSIFICATION MODEL FOR CYBERBULLYING DETECTION IN SOCIAL MEDIA

A MAJOR PROJECT REPORT

Submitted by

VUDITYALA SRINIDH **(Regd.No.18891A0558)**

Under the guidance of

Dr. G. Raja Vikram

Professor &

Head of the Department, CSE

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE
Near Ramoji FIlm City, Deshmukhi (V), Yadadri Bhuvanagiri Dist., Telangana - 508 284.
Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad
EAMCET CODE : VGNT PGECET CODE : VGNT1

JULY, 2022

DECLARATION

We hereby declare that project entitled "**An Efficient Classification Model for Cyberbullying Detection in Social Media**" is bonafide work duly completed by us. It does not contain any part of the project or thesis submitted by any other candidate to this or any other institute of the university.

All such materials that have been obtained from other sources have been duly acknowledged.

VUDITYALA SRINIDH
(Regd.No.18891A0558)

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE
Near Ramoji FIlm City, Deshmukhi (V), Yadadri Bhuvanagiri Dist., Telangana - 508 284.
Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad
EAMCET CODE : VGNT PGECET CODE : VGNT1

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the thesis work titled " **An Efficient Classification Model for Cyberbullying Detection in Social Media** " submitted by Mr. Vudityala Srinidh in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** to the Vignan Institute of Technology And Science, Deshmukhi is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted in any university for the award of any degree and the results are achieved satisfactorily.

Dr. G. Raja Vikram	**Dr. G. Raja Vikram**
Professor &	**Professor &**
Head of the Department, CSE	**Head of the Department, CSE**
GUIDE	

External Examiner

ACKNOWLEDGEMENT

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. We sincerely appreciate the inspiration; support and guidance of all those people who have been instrumental in making this project a success.

We thank our beloved **Chairman, Dr. L Rathaiah** Sir, who gave us great encouragement to work.

We thank our beloved **CEO**, **Mr. Boyapati Shravan** Sir, we remember him for his valuable ideas and facilities available in college during the development of the project.

We convey our sincere thanks to **Dr. G Durga Sukumar** Sir**, Principal** of our institution for providing us with the required infrastructure and a very vibrant and supportive staff.

We would like to thank our Head of the Department of Computer Science And Engineering and our guide of the project **Dr. G. Raja Vikram**, a distinguished and eminent personality, whose strong recommendation, immense support and constant encouragement has been great help to us. We intensely thank him for the same.

We would like to express our sincere appreciations to our project coordinators **Mr. N. Sri Anjaneya (Associate Professor) & Dr. O. Sri Nagesh (Professor)** for their guidance, continuous encouragement and support during the project.

Special thanks go to my team mates, who helped me to assemble the parts and gave suggestions in making this project. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices. We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our career.

ABSTRACT

From the day internet came into existence, the era of social networking sprouted. In the beginning, no one may have thought internet would be a host of numerous amazing services like the social networking. Today we can say that online applications and social networking websites have become a non-separable part of one's life. Many people from diverse age groups spend hours daily on such websites. Despite the fact that people are emotionally connected together through social media, these facilities bring along big threats with them such as cyber-attacks, which includes cyberbullying. As social networking sites are increasing, cyber bullying is increasing day by day. To identify word similarities in the tweets made by bullies and make use of machine learning and can develop an ML model automatically detect social media bullying actions. However, many social media bullying detection techniques have been implemented, but many of them were textual based. Under this background and motivation, it can help to prevent the happen of cyberbullying if we can develop relevant techniques to discover cyberbullying in social media. A machine learning model such as linear regression and stochastic gradient is proposed to detect and prevent bullying on Twitter.

Table of Contents

LIST OF FIGURES

Chapter 1

INTRODUCTION

Social networking sites are being widely used today for multiple purposes like entertainment, networking, etc. Social networking sites are a stop for multiple reasons to billions of people today. All the social media platforms require the consent of all the participating people. Communicating with people is no exception, as technology has changed the way people interact with a broader manner and has given a new dimension to communication. Many people are illegally using these communities. Many youngsters are getting bullied these days. Bullies use various services like Twitter, Facebook, and Email to bully people.

Cyberbullying is one of the most frequently happen Internet abuse and also a very serious social problem especially for teenager. Therefore, more and more researchers are devoting on how to discover and prevent the happen of cyberbullying, especially in social media. Cyberbullying is not just limited to creating a fake identity and publishing/posting some embarrassing photo or video, unpleasant rumours about someone but also giving them threats. The impacts of cyberbullying on social media are horrifying, sometimes leading to the death of some unfortunate victims.

What is Cyberbullying?

In today's world which has been made smaller by technology, new age problems have been born. No doubt technology has a lot of benefits; however, it also comes with a negative side. It has given birth to cyberbullying. To put it simply, cyberbullying refers to the misuse of information technology with the intention to harass others.

Subsequently, cyberbullying comes in various forms. It doesn't necessarily mean hacking someone's profiles or posing to be someone else. It also includes posting negative comments about somebody or spreading rumors to defame someone. As everyone is caught up on the social network, it makes it very easy for anyone to misuse this access.

In other words, cyberbullying has become very common nowadays. It includes actions to manipulate, harass and defame any person. These hostile actions are seriously damaging and can affect anyone easily and gravely. They take place on social media, public forums, and other online information websites. A cyberbully is not necessarily a stranger; it may also be someone you know.

Thus, a complete solution is required for this problem. Cyberbullying needs to stop. The problem can be tackled by detecting and preventing it by using a machine learning approach, this needs to be done using a different perspective.

Due to the significant development of Internet 2.0 technology, social media sites such as Twitter and Facebook have become popular and play a significant role in transforming human life. In particular, social media networks have incorporated daily activities, such as education, business, entertainment, and e-government, into human life. According to, social networking impacts are projected to exceed 3.02 billion active social media users each month globally by 2021. This number will account for approximately one-third of the Earth's population. Moreover, among the numerous existing social networks, Twitter is a critical platform and a vital data source for researchers. Twitter is a popular public microblogging network operating in real-time, in which news often appears before it appears in official sources. Characterized by its short message limit (now 280 characters) and unfiltered feed, Twitter use has rapidly increased, with an average of 500 million tweets posted daily, particularly during events . Currently, social media is an integral element of daily life. Undoubtedly, however, young people's usage of technology, including social media, may expose them to many behavioural and psychological risks. One of these risks is cyberbullying, which is an influential social attack occurring on social media platforms. In addition, cyberbullying has been associated with adverse mental health effects, including depression, anxiety, and other types of self-harm, suicidal thoughts, attempted suicide, and social and emotional difficulties.

Furthermore, the substantial increase in the number of cyberbullying cases has highlighted the danger of cyberbullying, particularly among children and adolescents, who can be inconsiderate and juvenile. According to several studies have shown that bullies often suffer from psychological conditions, leading them to bully and inflict suffering on others. Thus, cyberbullying is similar to an epidemic, and can lead to an aggressive society, particularly regarding high-tech university and school students.

1.1 Purpose

The main purpose is,

To provide an innovative preventative strategy that enables **fast and accurate detection** of cyberbullying case. Using ML techniques to predict cyberbullying produces high performance. To effectively **classify and predict** the data. To decrease **sparsity** problem. To enhance the performance of the overall prediction results.

1.2 Study of Existing System

The advent of social media, particularly Twitter, raises many issues due to a misunderstanding regarding the concept of freedom of speech. One of these issues is cyberbullying, which is a critical global issue that affects both individual victims and societies. Many attempts have been introduced in the literature to intervene in, prevent, or mitigate cyberbullying; however, because these attempts rely on the victims' interactions, they are practical. Therefore, detection of cyberbullying without the involvement of the victims is necessary. In this study, we attempted to explore this issue by compiling a global dataset of 37,373 unique tweets from Twitter. Moreover, seven machine learning classifiers were used, namely, Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), Random Forest (RF), AdaBoost (ADB), Naive Bayes (NB), and Support Vector Machine (SVM).In existing, Cyberbullying incidents are increasing day by day as technology rolls out. A large number of cyberbullying incidents are reported by companies each year. The existing system doesn't effectively classify and predict the tweets which is presented in the social media.

Disadvantages :

- Doesn't Efficient for handling large volume of data.
- Theoretical Limits
- Incorrect Classification Results.
- Less Prediction Accuracy.

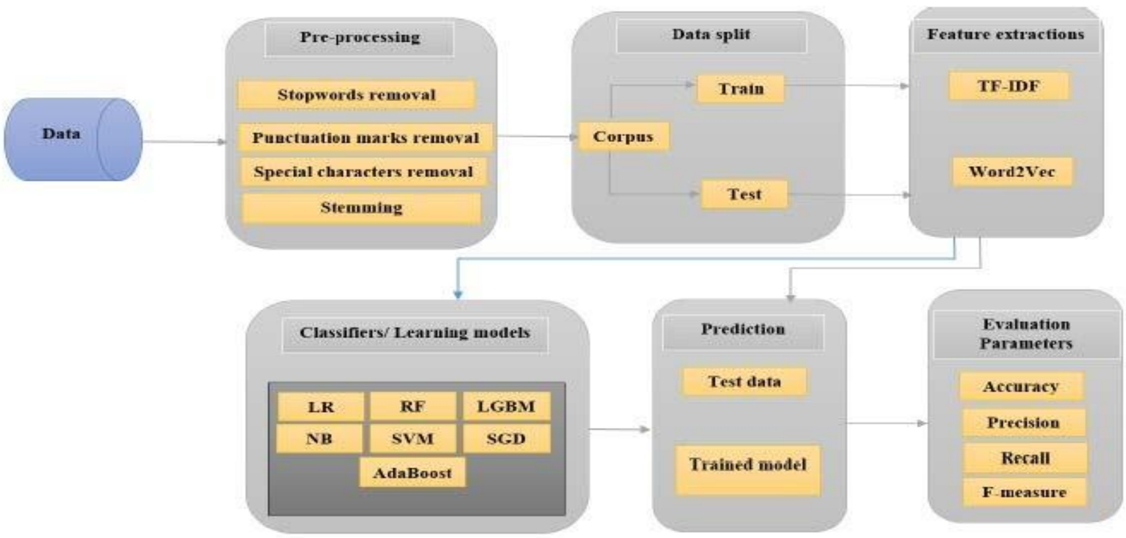

FIGURE 1.1: EXISTING SYSTEM MODEL

1.3 Proposed System

The proposed model is introduced to overcome all the disadvantages that arises in the existing system. In this system, we have to take the twitter cyberbullying dataset as input. Then we have to implement the Natural Language Processing (NLP) techniques for text cleaning. Then we have to implement the feature extraction technique for improving the existing system results. Here, we have to improve the detection accuracy rate by using the feature extraction such as count vectorization. It means, to encode the text as integers or numeric value to create the feature vectors. After that, we have to implement the two different machine learning algorithms such as linear regression and stochastic gradient descent algorithm. The experimental results shows that, some performance metrics such as accuracy. In proposed system, we have to improve the accuracy rate when compared with existing system.

Advantages:

- It is efficient for large number of datasets.
- To implement the feature extraction technique.
- The experimental result is high when compared with existing system.
- The prediction results is efficient.
- To classify the result effectively.
- Time consumption is low.

Chapter 2

REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS:

- O/S : Windows 7.
- Language : Python
- Front End : Anaconda Navigator – Spyder

Python:

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

Features of Python:

- **Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

- **Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

- **Free and Open Source**

Python is an example of a *FLOSS* (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

- **High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

- **Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like Kivy to create games for your computer *and* for iPhone, iPad, and Android.

- **Interpreted**

This requires a bit of explanation.

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just *run* the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc.

This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

- **Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In *procedure-oriented* languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In *object-oriented* languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

- **Extensible**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

- **Embeddable**

You can embed Python within your C/C++ programs to give *scripting* capabilities for your program's users.

- **Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the *Batteries Included* philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface included in Anaconda that allows you to launch applications and easily manage conda packages, environments and channels without the need to use command line commands.

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

Spyder:

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features.

2.2 HARDWARE REQUIREMENTS:

- System: Pentium IV 2.4 GHz
- Hard Disk: 200 GB
- Mouse: Logitech.
- Keyboard: 110 keys enhanced
- Ram: 4GB

Chapter 3

LITERATURE SURVEY

3.1 Title: Detecting Offensive Language in Social Media to Protect Adolescent Online Safety.

Year: 2012

Author: Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu

Methodology: user-level offensiveness detection seems a more feasible approach. so, the Lexical Syntactic Feature (LSF) architecture to detect offensive content and identify potential offensive users in social media. We distinguish the contribution of pejoratives/profanities and obscenities in determining offensive content, and introduce hand-authoring syntactic rules in identifying name-calling harassments. In particular, we incorporate a user's writing style, structure and specific cyberbullying content as features to predict the user's potentiality to send out offensive content.

Advantage

- Lexical Syntactic Feature (LSF) detection is fast and accurate.

Disadvantage

- Lexical Syntactic Feature (LSF) cannot handle large data for prediction.

3.2 Title: Opinion Mining and Social Networks: a Promising Match

Year: 2011

Author: K. Jedrzejewski and M. Morzy.

Methodology: The role and importance of social networks in preferred environments for opinion mining and sentiment analysis. Selected properties of social networks that are relevant with respect to opinion mining are described and general relationships between the two disciplines are outlined. The related work and basic definitions used in opinion mining is given. Then, our original method of opinion classification is introduced and we test the algorithm on datasets acquired from social networks and thus report the results.

Advantage

- Sentiment analysis is a useful for any organization for which public sentiment or attitude towards them is important for their success

Disadvantage

- Automated sentiment analysis tools do a really great job of analysing text for opinion and attitude, but they're not perfect.

3.3 Title: Analyzing Labelled Cyberbullying Incidents on the Instagram Social Network.

Year: 2015

Author: H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra.

Methodology: Cyberbullying is a growing problem affecting more than half teens. The main goal is to study cyberbullying incidents in the social network. In this work, we have collected a sample data and their associated comments. We then designed a study and employed human contributors at the crowd-sourced Crowd Flower website to label these media sessions for cyberbullying. A detailed analysis of the labelled data is then presented, including a study of relationships between cyberbullying and a host of features.

Advantage

- Crowd Flower transcribes data from multiple sources into comprehensible transcripts.

Disadvantage

- Crowd Flower workforce has limitations and are difficult to manage.

3.4 Title: Using Machine Learning to Detect Cyberbullying

Year: 2011

Author: Kelly Reynolds, April Kontostathis, Lynne Edwards

Methodology: Cyberbullying is the use of technology as a medium to bully someone. It has been an issue for many years, the recognition of its impact on young people has recently increased. Social networking sites provide a fertile medium for bullies, and teens and young adults who use these sites are vulnerable to attacks. Through machine learning, we can detect language patterns used by bullies and their victims, and develop rules to automatically detect cyberbullying content.

Advantage

- With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own

Disadvantage

- Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality there can also be times where they must wait for new data to be generated

CHAPTER 4

SYSTEM DIAGRAMS

4.1 SYSTEM ARCHITECTURE:

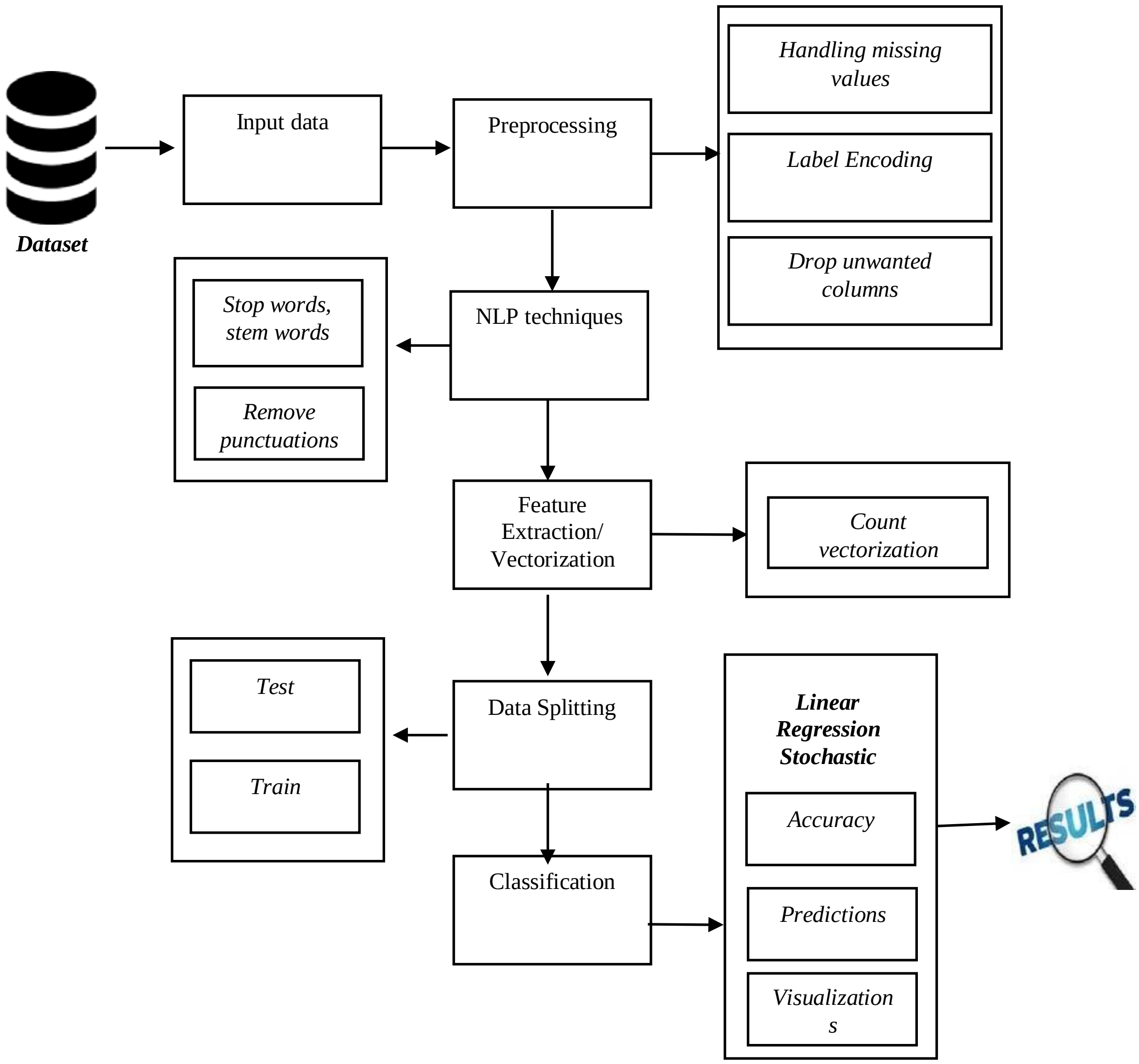

FIGURE 4.1: SYSTEM ARCHITECTURE

A **system architecture** is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

4.2 FLOW DIAGRAM:

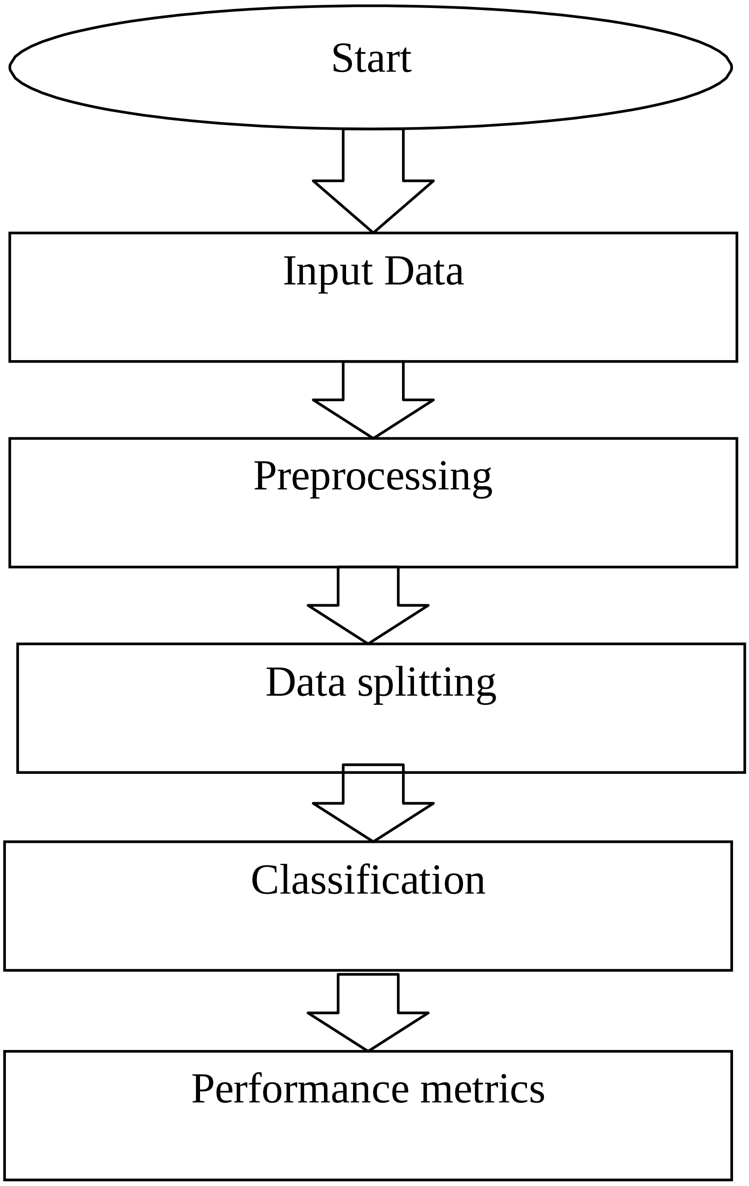

FIGURE 4.2: FLOW DIAGRAM

Flow diagram is a collective term for a diagram representing a flow or set of dynamic relationships in a system. The term flow diagram is also used as a synonym for flowchart, and sometimes as a counterpart of the flowchart.

Flow diagrams are used to structure and order a complex system, or to reveal the underlying structure of the elements and their interaction.

4.3 CLASS DIAGRAM:

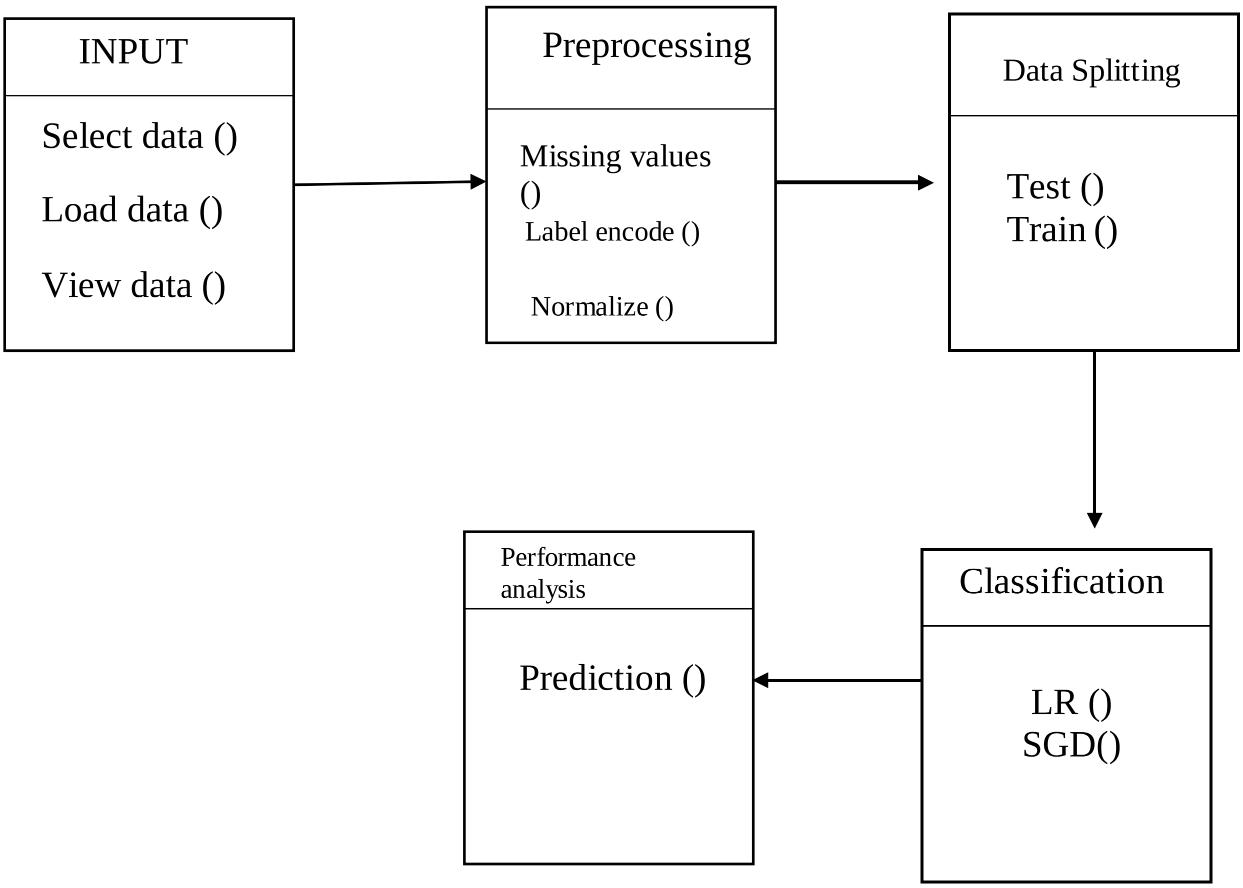

FIGURE 4.3: CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

4.4 USE CASE DIAGRAM:

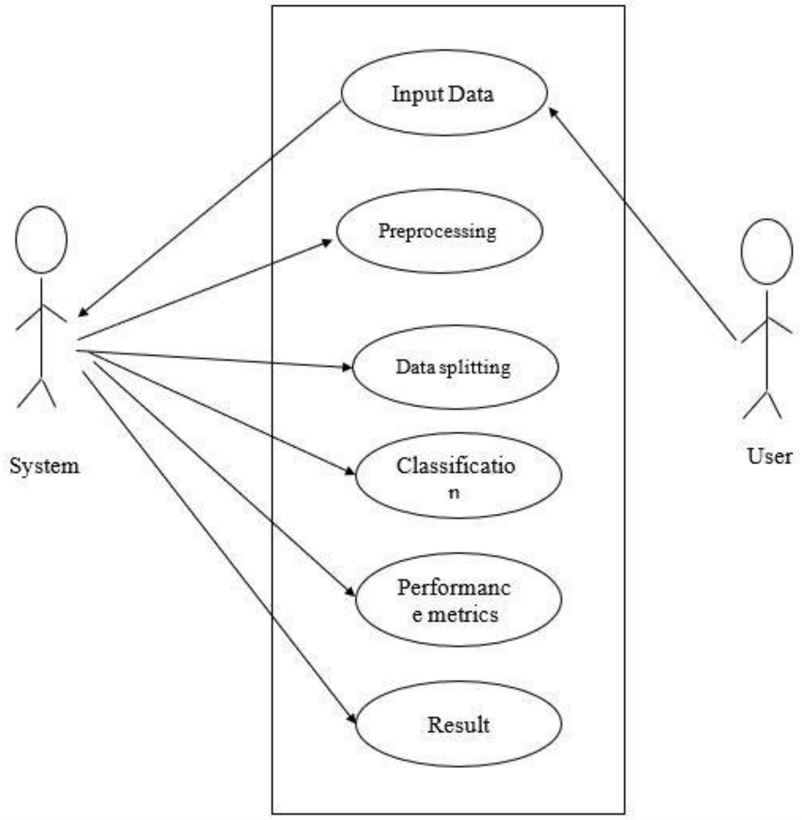

FIGURE 4.4: USE CASE DIAGRAM

A **use case diagram** is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system".

4.5 ACTIVITY DIAGRAM:

An activity diagram is a **behavioral diagram** i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram.

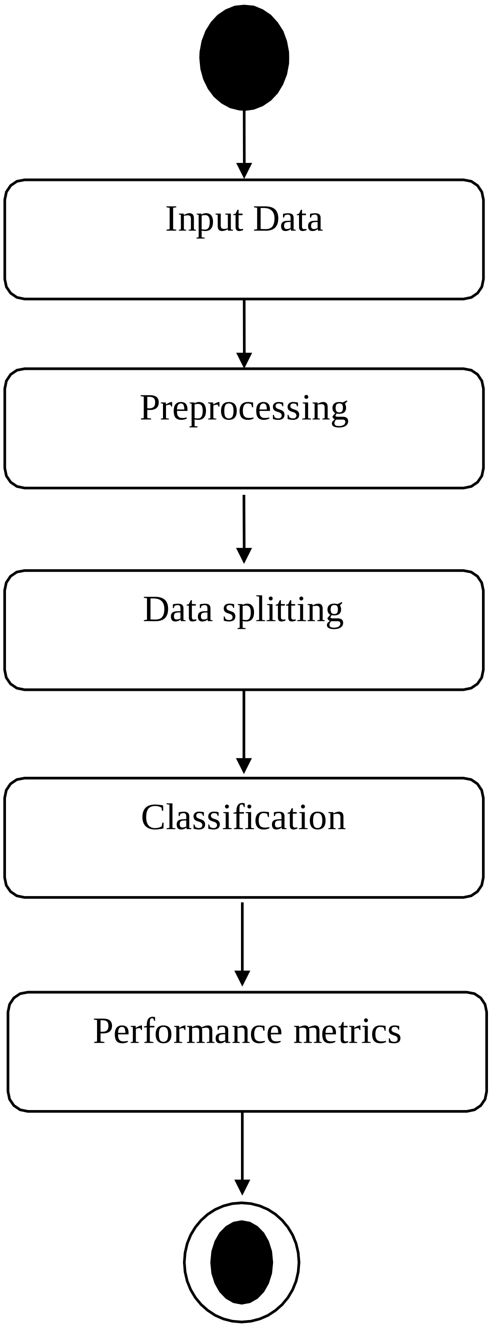

FIGURE 4.5: ACTIVITY DIAGRAM

4.6 SEQUENCE DIAGRAM:

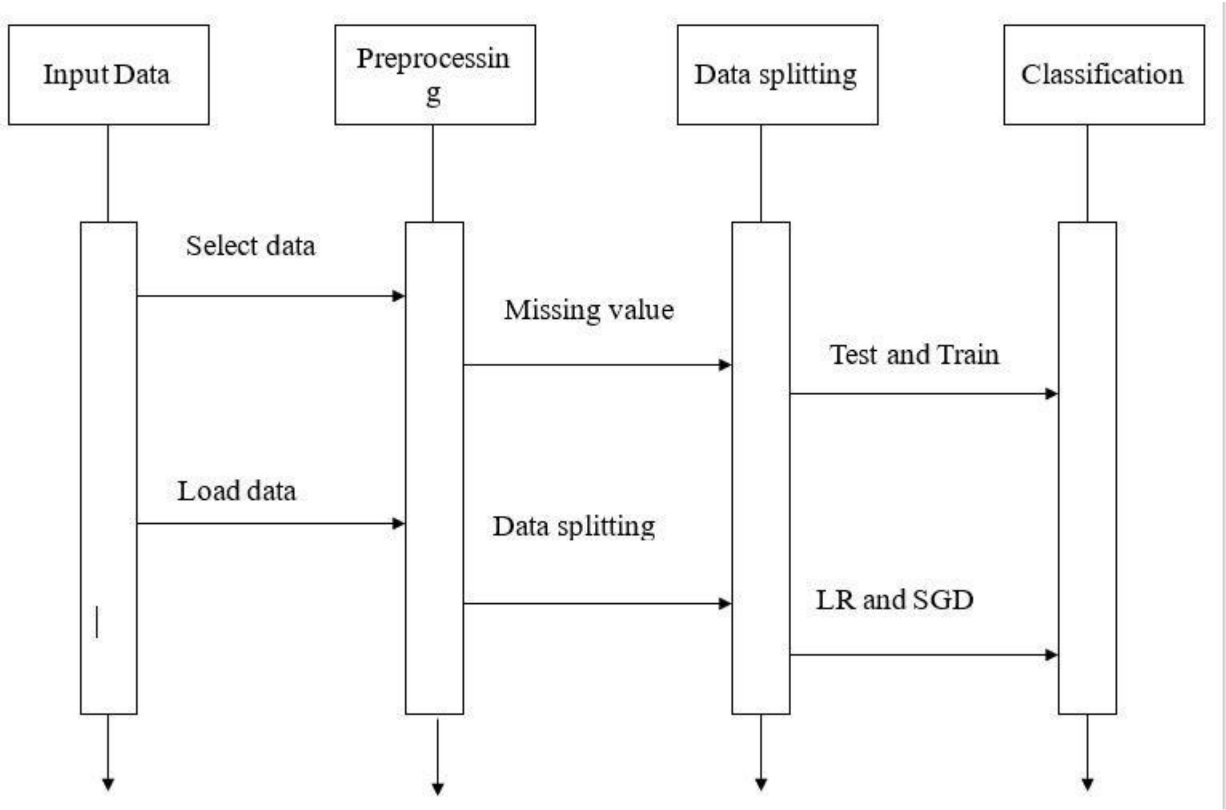

FIGURE 4.6: SEQUENCE DIAGRAM

A **sequence diagram** or **system sequence diagram** (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

4.7 ER DIAGRAM:

An **entity relationship** diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

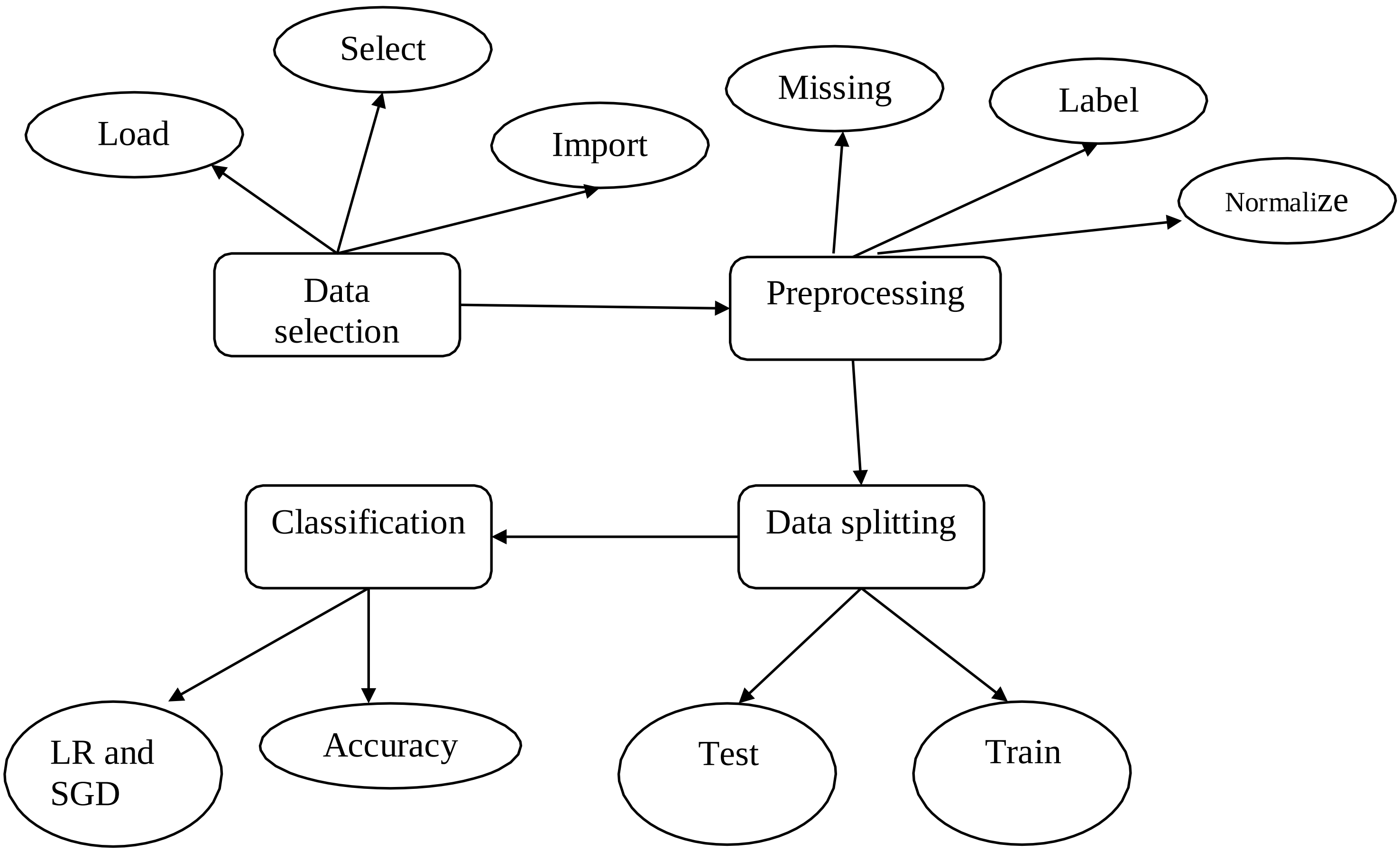

FIGURE 4.7: ER DIAGRAM

CHAPTER 5

IMPLEMENTATION

Cyberbullying:

When a young person uses the Internet or technology to harass, threaten, embarrass, or target another person, this person is called a cyberbully. Typically, cyberbullying involves tweens and teens; but it's not uncommon for adults to experience cyberbullying and public shaming as well.

Compared to traditional bullying, the effects of cyberbullying are often more significant. Not only do the hurtful messages reach an unlimited audience, but the words and images are often preserved online.

Even if someone deletes a mean post, chances are it's still available in some form such as in a screenshot or a shared text message. Worse yet, those who are targeted by cyberbullies often don't know who is bullying them, so they often have no way to bring it to an end.

Types of Cyberbullying:

Kids are online now more than ever.

Every day they use their smartphones, tablets, and computers not only to research material for school but to socialize with friends and family members. In fact, texting and using social media is one of the top ways kids communicate with others.

But just like any other social activity, the opportunity for bullying exists. While there are a number of ways kids bully others online, the majority of online harassment falls into one of five categories. These include harassment, impersonation, photograph use, website creation, and video shaming. Here is what you need to know about the most common types of cyberbullying.

Harassment: Harassing someone is a common method of online bullying. This type of cyberbullying occurs when someone uses technology to torment another person. One way kids harass others is by engaging in warning wars. This occurs when they use a site's report button as a way to get another person in trouble or kicked offline—even when they are doing nothing wrong.

Impersonation: Another common form of cyberbullying is impersonation, where one person impersonates another person online. Although there are a number of ways for kids to accomplish this, one of the most common is to hack the account or steal the password and make changes to the target's profile.

Once they have access, they might post sexual, racist, or other inappropriate things to ruin the target's social standing and reputation. Or, they might chat with other people while pretending to be the victim. They will say mean things with the purpose of offending and angering the target's friends or acquaintances.

Inappropriate Photographs: People who cyberbully others will sometimes use photographs to bully or shame other people. These photos may include embarrassing or inappropriate images that were either shared privately with them or were taken without the target knowing like in a locker room, a bathroom, or dressing room.

They then use these photos as weapons and post them on social media or on photo-sharing sites for anyone on the internet to view and download. Other times, they might send mass emails or text messages that include nude or degrading photos of the target.

Website Creation: Sometimes, kids who cyberbully others will create a website, blog, or poll to harass another person. For instance, they might conduct an internet poll about a target or several targets. Questions in the poll may include extremely hurtful questions like asking people to rank their peers by their looks or their weight.

Impact of Cyberbullying:

When cyberbullying occurs, kids experience a variety of physical, psychological, and emotional consequences. They may complain of everything from fear and anxiety to depression and low self-esteem. They also may struggle academically and report feelings of significant distress. In fact, more than 30% of kids who are targeted by cyberbullies report experiencing symptoms of stress.

Victims of cyberbullying also find it difficult to feel safe and may feel alone and isolated, especially if they are being ostracized by their peers. Cyberbullying can lead to increasing levels of anxiety and depression. One study found that as many as 93% of kids victimized by cyberbullies reported feelings of sadness, hopelessness, and powerlessness.

Twitter is one of the platforms where cberbullying is happening, so now let us see what is twitter.

Twitter:

Twitter is a microblogging and social networking service on which users post and interact with messages known as "tweets", owned by American company Twitter, Inc. Registered users can post, like, and retweet tweets, however, unregistered users have the ability to only read tweets that are publicly available. Users interact with Twitter through browser or mobile frontend software, or programmatically via its APIs. Prior to April 2020, services were accessible via SMS. Tweets were originally restricted to 140 characters, but the limit was doubled to 280 for non-CJK languages in November 2017. Audio and video tweets remain limited to 140 seconds for most accounts.

Tweets are publicly visible by default, but senders can restrict message delivery to only their followers. Users can mute users they do not wish to interact with, block accounts from viewing their tweets and remove accounts from their followers list.[136][137][138] Users can tweet via the Twitter website, compatible external applications (such as for smartphones), or by Short Message Service (SMS) available in certain countries. Users may subscribe to other users' tweets—this is known as "following" and subscribers are known as "followers" or "tweeps", a portmanteau of Twitter and peeps. Individual tweets can be forwarded by other users to their own feed, a process known as a "retweet". In 2015, Twitter launched "quote tweet" (originally called "retweet with comment"), a feature that allows users to add a comment to their retweet, nesting one tweet in the other. Users can also "like" (formerly "favourite") individual tweets.

The counters for "likes", "retweets", and replies appear next to the respective buttons in timelines such as on profile pages and search results. Counters for likes and retweets exist on a tweet's standalone page too. Since September 2020, quote tweets, formerly known as "retweet with comment", have an own counter on their tweet page.[142] Until the legacy desktop front end that was discontinued in 2020, a row with miniature profile pictures of up to ten liking or retweeting users was displayed (earliest documented implementation in December 2011 overhaul), as well as a tweet reply counter next to the according button on a tweet's page.

So it is important to detect these cyberbullying tweets from these social media platforms and protect everyone from danger.

Now let us see what are the modules present in the process of detecting cyberbullying tweets in the twitter dataset.

MODULES:

- Data Selection
- Data Preprocessing
- NLP Techniques
- Data splitting
- Feature Extraction
- Classification
- Performance metrics
- Prediction

5.1 DATA SELECTION:

Data selection is defined as the process of determining the appropriate **data type** and **source**, as well as suitable **instruments** to collect data. Data selection precedes the actual practice of data collection.

The process of selecting suitable data for a research project can impact data integrity.

The primary objective of data selection is the determination of appropriate data type, source, and instrument(s) that allow investigators to adequately answer research questions. This determination is often discipline-specific and is primarily driven by the nature of the investigation, existing literature, and **accessibility** to necessary data sources.

The input data was collected from dataset repository. In this project, the cyberbullying tweets dataset is used for detecting offensive and non-offensive tweets. The dataset which contains the information about the user name and tweets label.

5.2 DATA PREPROCESSING:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. Data pre-processing is the process of removing the unwanted data from the dataset. Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning. This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient. Missing data removal. Encoding Categorical data. Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0. Missing and duplicate values were removed and data was cleaned of any abnormalities. Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.

That most machine learning algorithms require numerical input and output variables.

5.3 NLP TECHNIQUES:

NLP is a field in machine learning with the ability of a computer to understand, analyze, manipulate, and potentially generate human language. Cleaning (or pre-processing) the data typically consists of a number of steps: *Remove punctuation*: Punctuation can provide grammatical context to a sentence which supports our understanding. *Tokenization*: Tokenizing separates text into units such as sentences or words. It gives structure to previously unstructured text. eg: Plata o Plomo-> ‘Plata’,’o’,’Plomo’. *Stemming*: Stemming helps reduce a word to its stem form.

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

Sentiment analysis focuses on the polarity of a text (*positive, negative, neutral*) but it also goes beyond polarity to detect specific feelings and emotions (*angry, happy, sad*, etc), urgency (*urgent, not urgent*) and even intentions (*interested, not interested*).

Here we use VADER sentiment analyser to analyse the sentiment of the tweet.

VADER Sentiment Analysis:

VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.

VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

positive sentiment: (compound score >= 0.05)

neutral sentiment: (compound score > -0.05) and (compound score < 0.05)

negative sentiment: (compound score <= -0.05)

VADER sentiment analysis combines a dictionary of lexical features to sentiment scores with a set of five heuristics. The model works best when applied to social media text, but it has also proven itself to be a great tool when analyzing the sentiment of movie reviews and opinion articles.

Five Simple Heuristics

Lexical features aren't the only things in the sentence which affect the sentiment. There are other contextual elements, like punctuation, capitalization, and modifiers which also impart emotion. VADER sentiment analysis takes these into account by considering five simple heuristics. The effect of these heuristics are, again, quantified using human raters.

- The first heuristic is **punctuation**. Compare "I like it." and "I like it!!!" It's not really hard to argue that the second sentence has more intense emotion than the first, and therefore must have a higher VADER sentiment score.
- The second heuristic is **capitalization**. "AMAZING performance." is definitely more intense than "amazing performance." And so VADER takes this into account by incrementing or decrementing the sentiment score of the word by 0.733, depending on whether the word is positive or negative, respectively.
- The third heuristic is the use of **degree modifiers**. Take for example "effing cute" and "sort of cute". The effect of the modifier in the first sentence is to increase the intensity of cute, while in the second sentence, it is to decrease the intensity. VADER maintains a booster dictionary which contains a set of boosters and dampeners.
- The fourth heuristic is the **shift in polarity due to "but"**. Oftentimes, "but" connects two clauses with contrasting sentiments. The dominant sentiment, however, is the latter one. For example, "I love you, but I don't want to be with you anymore." The first clause "I love you" is positive, but the second one "I don't want to be with you anymore." is negative and obviously more dominant sentiment-wise, here VADER implements a "but" checker.

- The fifth heuristic is **examining the tri-gram before a sentiment-laden lexical feature to catch polarity negation**. Here, a tri-gram refers to a set of three lexical features. VADER maintains a list of negator words. Negation is captured by multiplying the sentiment score of the sentiment-laden lexical feature by an empirically-determined value -0.74.

The sentiment score of a sentence is calculated by summing up the sentiment scores of each VADER-dictionary-listed word in the sentence. Cautious readers would probably notice that there is a contradiction: individual words have a sentiment score between -4 to 4, but the returned sentiment score of a sentence is between -1 to 1.

They're both true. The sentiment score of a sentence is the sum of the sentiment score of each sentiment-bearing word. However, we apply a normalization to the total to map it to a value between -1 to 1.

The normalization used by Hutto is

$$\frac{x}{\sqrt{x^2 + \alpha}}$$

where x is the sum of the sentiment scores of the constituent words of the sentence and alpha is a normalization parameter that we set to 15. The normalization is graphed below.

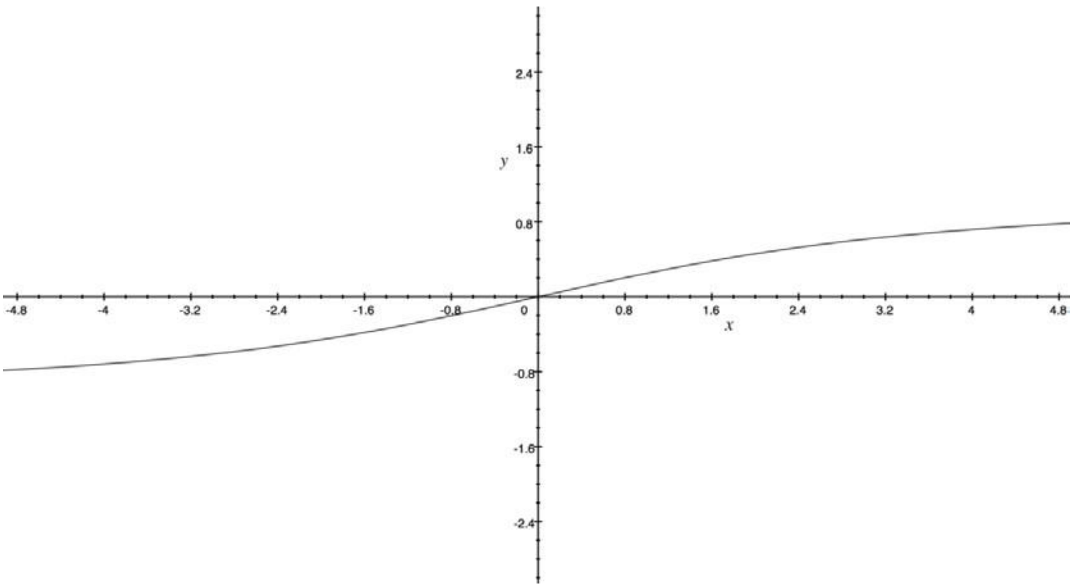

FIGURE 5.1: NORMALIZATION GRAPH

We see here that as x grows larger, it gets more and more close to -1 or 1. To similar effect, if there are a lot of words in the document you're applying VADER sentiment analysis to, you get a score close to -1 or 1. Thus, VADER sentiment analysis works best on short documents, like tweets and sentences, not on large documents.

5.4 DATA SPLITTING:

Data is at the heart of every ML problem. Without proper data, ML models are just like bodies without soul. But in today's world of 'big data' collecting data is not a major problem anymore. We are knowingly (or unknowingly) generating huge datasets every day. However, having surplus data at hand still does not solve the problem. For ML models to give reasonable results, we not only need to feed in large quantities of data but also have to ensure the quality of data.

Though making sense out of raw data is an art in itself and requires good feature engineering skills and domain knowledge (in special cases), the quality data is of no use until it is properly used. The major problem which ML/DL practitioners face is how to divide the data for training and testing. Though it seems like a simple problem at first, its complexity can be gauged only by diving deep into it. Poor training and testing sets can lead to unpredictable effects on the output of the model. It may lead to overfitting or underfitting of the data and our model may end up giving biased results.

Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.

- **Train Set:**

  The train set would contain the data which will be fed into the model. In simple terms, our model would learn from this data. For instance, a Regression model would use the examples in this data to find gradients in order to reduce the cost function. Then these gradients will be used to reduce the cost and predict data effectively.

- **Test Set:**

  The test set contains the data on which we test the trained and validated model. It tells us how efficient our overall model is and how likely is it going to predict something which does not make sense. There are a plethora of evaluation metrics (like precision, recall, accuracy, etc.) which can be used to measure the performance of our model.

In our process, we considered 70% of the dataset to be the training data and the remaining 30% to be the testing data.

5.5 FEATURE EXTRACTION:

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. Feature extraction can be accomplished manually or automatically:

- Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful. Over decades of research, engineers and scientists have developed feature extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.
- Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarised version of the original features can be created from a combination of the original set.

In this we are using the count vectorization technique in the feature extraction process.

Count Vectorization:

In order to use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called **tokenization**. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called **feature extraction (or vectorization)**.

Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

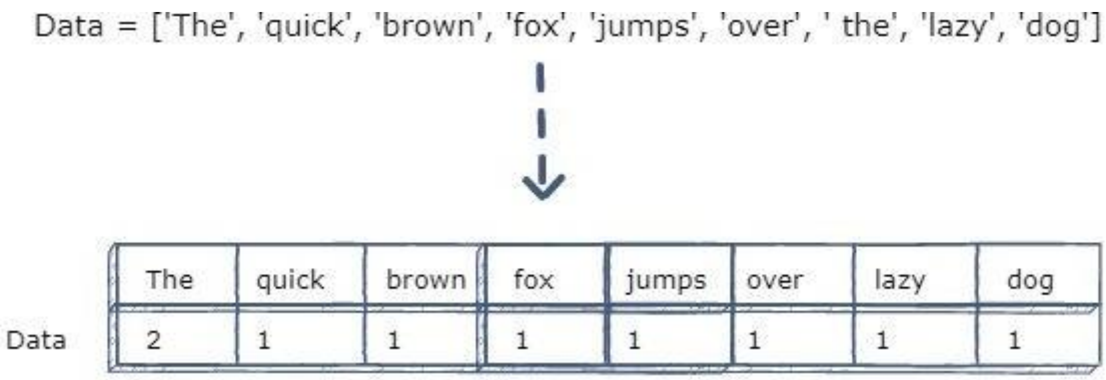

FIGURE 5.2: COUNT VECTORIZER

5.6 CLASSIFICATION:

A common job of machine learning algorithms is to recognize objects and being able to separate them into categories. This process is called classification, and it helps us segregate vast quantities of data into discrete values, i.e.: distinct, like 0/1, True/False, or a pre-defined output label class.

Classification is defined as the process of recognition, understanding, and grouping of objects and ideas into preset categories a.k.a "sub-populations." With the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future datasets into respective and relevant categories.

Classification algorithms used in machine learning utilize input training data for the purpose of predicting the likelihood or probability that the data that follows will fall into one of the predetermined categories. One of the most common applications of classification is for filtering emails into "spam" or "non-spam", as used by today's top email service providers.

In short, classification is a form of "pattern recognition,". Here, classification algorithms applied to the training data find the same pattern (similar number sequences, words or sentiments, and the like) in future data sets.

In this step, we have to implement the two different machine learning algorithms such as ***Stochastic gradient descent and Linear regression.*** With the help of machine learning algorithms, we have to analyse the cyberbullying cases.

Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

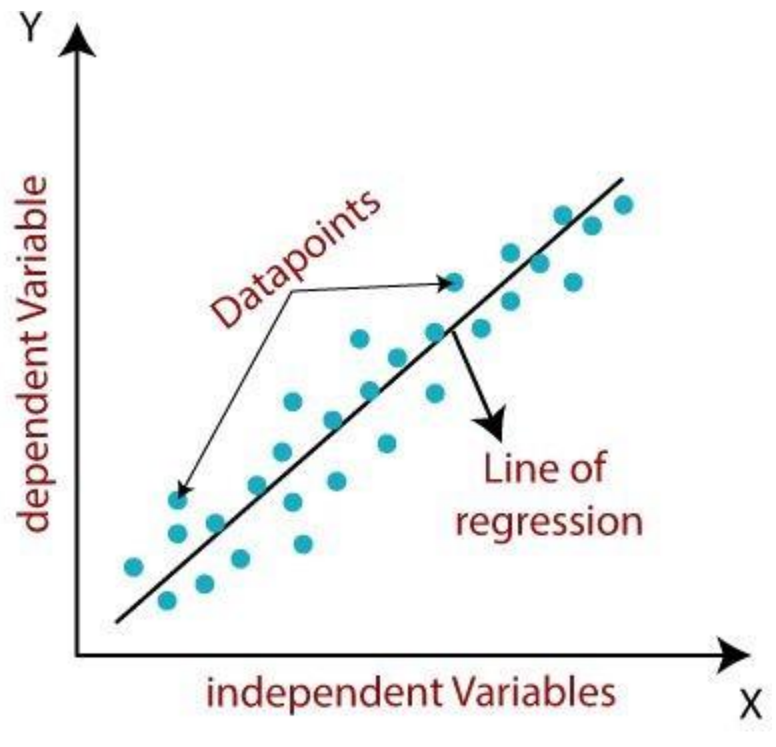

FIGURE 5.3: LINEAR REGRESSION

Linear regression can be further divided into two types of the algorithm:

- **Simple Linear Regression:**

  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

- **Multiple Linear regression:**

  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

When Do You Need Regression?

Typically, you need regression to answer whether and how some phenomenon influences the other or how **several** variables are related. For example, you can use it to determine *if* and *to what extent* experience or gender impacts salaries.

Regression is also useful when you want to **forecast** a response using a new set of predictors. For example, you could try to predict electricity consumption of a household for the next hour given the outdoor temperature, time of day, and number of residents in that household.

Regression is used in many different fields, including economics, computer science, and the social sciences. Its importance rises every day with the availability of large amounts of data and increased awareness of the practical value of data.

Steps to Implement Linear Regression:

Step 1: Import packages and classes

The first step is to import the package numpy and the class LinearRegression from sklearn.linear_model

Step 2: Provide data

The second step is defining data to work with. The inputs (regressors, x) and output (response, y) should be arrays or similar objects. This is the simplest way of providing data for regression.

Step 3: Create a model and fit it

The next step is to create a linear regression model and fit it using the existing data.

Create an instance of the class LinearRegression, which will represent the regression model.

Step 4: Get results

Once you have your model fitted, you can get the results to check whether the model works satisfactorily and to interpret it.

Step 5: Predict response

Once you have a satisfactory model, then you can use it for predictions with either existing or new data. To obtain the predicted response, use .predict()

Stochastic Gradient Descent:

Stochastic gradient descent is an optimization algorithm often used in machine learning applications to find the model parameters that correspond to the best fit between predicted and actual outputs. It's an inexact but powerful technique.

Stochastic gradient descent is widely used in machine learning applications. Combined with backpropagation, it's dominant in neural network training applications.

It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the very high computational burden, achieving faster iterations in trade for a lower convergence rate.

Basic Gradient Descent Algorithm:

The gradient descent algorithm is an approximate and iterative method for mathematical optimization. You can use it to approach the minimum of any differentiable function. Although gradient descent sometimes gets stuck in a local minimum or a saddle point instead of finding the global minimum, it's widely used in practice. Data science and machine learning methods often apply it internally to optimize model parameters. For example, neural networks find weights and biases with gradient descent.

Gradient, in plain terms means slope or slant of a surface. So gradient descent literally means descending a slope to reach the lowest point on that surface. Let us imagine a two dimensional graph, such as a parabola in the figure below.

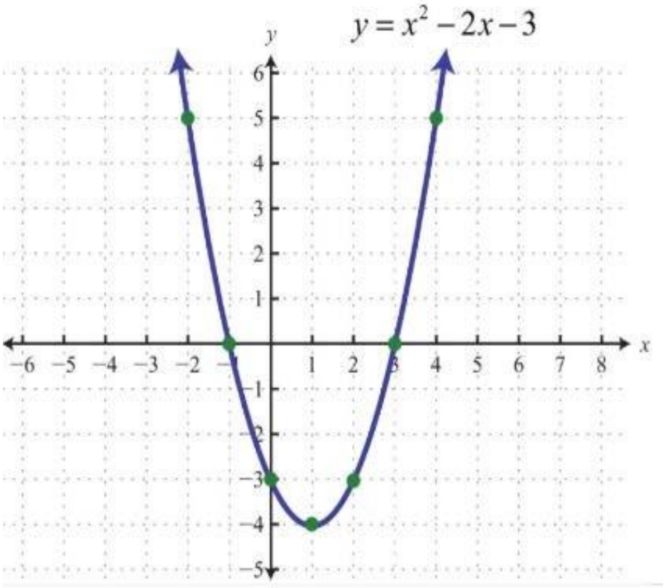

FIGURE 5.4: GRADIENT DESCENT

In the above graph, the lowest point on the parabola occurs at x = 1. The objective of gradient descent algorithm is to find the value of "x" such that "y" is minimum. "y" here is termed as the objective function that the gradient descent algorithm operates upon, to descend to the lowest point.

Why Stochastic Gradient Descent?

Say we have 10,000 data points and 10 features. The sum of squared residuals consists of as many terms as there are data points, so 10000 terms in our case. We need to compute the derivative of this function with respect to each of the features, so in effect we will be doing 10000 * 10 = 100,000 computations per iteration. It is common to take 1000 iterations, in effect we have 100,000 * 1000 = 100000000 computations to complete the algorithm. That is pretty much an overhead and hence gradient descent is slow on huge data.

So here, Stochastic gradient descent comes to our rescue!! "Stochastic", in plain terms means "random".

It is while selecting data points at each step to calculate the derivatives. SGD randomly picks one data point from the whole data set at each iteration to reduce the computations enormously.

It is also common to sample a small number of data points instead of just one point at each step and that is called "mini-batch" gradient descent. Mini-batch tries to strike a balance between the goodness of gradient descent and speed of SGD.

Steps to implement Stochastic Gradient Descent:

Step 1: First, we will import all the necessary libraries.

Step 2: Now, we will load our dataset. Here, X contains the dataset we have and Y contains the label that we need to predict.

Step 3: Now split your data before scaling to avoid the data leakage problem.

Step 4: Create the DataFrame using pandas.

Step 5: Implement SGD function.

5.7 PERFORMANCE METRICS:

Classification is a type of supervised machine learning problem where the goal is to predict, for one or more observations, the category or class they belong to.

An important element of any machine learning workflow is the evaluation of the performance of the model. This is the process where we use the trained model to make predictions on previously unseen, labelled data. In the case of classification, we then evaluate how many of these predictions the model got right.

In real-world classification problems, it is usually impossible for a model to be 100% correct. When evaluating a model it is, therefore, useful to know, not only how wrong the model was, but in which way the model was wrong. Some different performance metrics and techniques you can use to evaluate a classifier are:

1. Accuracy

The overall **accuracy** of a model is simply the number of correct predictions divided by the total

number of predictions. An accuracy score will give a value between 0 and 1, a value of 1 would indicate a perfect model.

AC= (TP+TN)/ (TP+TN+FP+FN)

2. Confusion Matrix

A **confusion matrix** is an extremely useful tool to observe in which way the model is wrong (or right!). It is a matrix that compares the number of predictions for each class that are correct and those that are incorrect.

In a confusion matrix, there are 4 numbers to pay attention to.

True positives: The number of positive observations the model correctly predicted as positive.

False-positive: The number of negative observations the model incorrectly predicted as positive.

True negative: The number of negative observations the model correctly predicted as negative.

False-negative: The number of positive observations the model incorrectly predicted as negative.

3. AUC/ROC

A classifier such logistic regression will return the probability of an observation belonging to a particular class as the prediction output. For the model to be useful this is usually converted to a binary value e.g. either the sample belongs to the class or it doesn't. To do this a classification threshold is used, for example, we might say that if the probability is above 0.5 then the sample belongs to class 1.

4. Precision

Precision measures how good the model is at correctly identifying the positive class. In other words out of all predictions for the positive class how many were actually correct? Using alone this metric for optimising a model we would be minimising the false positives. This might be desirable for our fraud detection example, but would be less useful for diagnosing cancer as we would have little understanding of positive observations that are missed.

Precision=TP/ (TP+FP)

5. Recall

Recall tell us how good the model is at correctly predicting **all** the positive observations in the dataset. However, it does not include information about the false positives so would be more useful in the cancer example.

Recall=TP/ (TP+FN)

6. F1 score

The **F1 score** is the harmonic mean of precision and recall. The F1 score will give a number between 0 and 1. If the F1 score is 1.0 this indicates perfect precision and recall. If the F1 score is 0 this means that either the precision or the recall is 0.

F1 = 2 * precision * recall / precision + recall

Assessing the performance of a classifier is generally not straightforward and highly dependant on the use case and available dataset. It is particularly important to understand the risk of being wrong in a particular direction so that you can produce a truly useful model.

5.8 PREDICTION:

A number is given as input in the runtime. The model takes the number and considers it as an index number. Then the tweet present at that index is predicted as Cyberbullying case or Non Cyberbullying case.

Some cases are shown in the below figures.

```
******************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case
```

FIGURE 5.5: INPUT

Prediction case 1

```
(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 16
```

FIGURE 5.6: CASE 1

Answer for the above input is shown in the below figure

```
*******************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 16

-- Non cyberbullying case --

***********************************
```

FIGURE 5.7: CASE 1 ANSWER

Prediction case 2

```
******************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 1952
```

FIGURE 5.8: CASE 2

The answer for above input is shown in the below figure

```
********************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 1952
***********************************

--Cyberbullying case --
```

FIGURE 5.9: CASE 2 ANSWER

CHAPTER 6

CODING

IMPORTING PACKAGES

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
import re
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import SGDClassifier
import warnings
warnings.filterwarnings
```

Pandas is used to analyze data. It is made mainly for working with relational or labeled data both easily and intuitively. Matplotlib is used to visualize data using a variety of different types of plots to make data understandable. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

The re module raises the exception re. error if an error occurs while compiling or using a regular expression. We would cover two important functions, which would be used to handle regular expressions.

READING DATA

```
data=pd.read_csv('Cyberbullying.csv')
print("********************************************")
print()
print("Data Selection")
print()
```

```
print("********************************************")
print(data.head(10))
print()
```

This part of code loads the dataset and prints the first 10 rows of the dataset.

DATA PREPROCESSING

```
#=== CHECK MISSING VALUES ===
print("********************************************")
print()
print(" Handling missing values")
print()
print(data.isnull().sum())
print()
#=== DROP UNWANTED COLUMNS ===
data=data.drop(['annotation'], axis = 1)
```

Here the missing values in the columns are handled by replacing them with 0 or mean value of that column and unwanted columns are removed.

NLP TECHNIQUE

```
#==== text cleaning ====
cleanup_re = re.compile('[^a-z]+')
def cleanup(sentence):
    sentence = str(sentence)
    sentence = sentence.lower()
    sentence = cleanup_re.sub(' ', sentence).strip()
    return sentence
data["Summary_Clean"] = data["content"].apply(cleanup)
print("********************************************")
print()
print("Before applying NLP techniques")
print()
print(data["content"].head(10))
print()
```

```
print("*********************************************")
print()
print("After applying NLP techniques")
print()
print(data["Summary_Clean"].head(10))
print()
```

Cleaning of data takes place, by converting text into lower case and stop words like comma, full stops etc are removed.

SENTIMENT ANALYSIS

```
analyzer = SentimentIntensityAnalyzer()
data['compound'] = [analyzer.polarity_scores(x)['compound'] for x in data['Summary_Clean']]
data['neg'] = [analyzer.polarity_scores(x)['neg'] for x in data['Summary_Clean']]
data['neu'] = [analyzer.polarity_scores(x)['neu'] for x in data['Summary_Clean']]
data['pos'] = [analyzer.polarity_scores(x)['pos'] for x in data['Summary_Clean']]
#=== Labelling ===
data['comp_score'] = data['compound'].apply(lambda c: 0 if c >=0 else 1)
```

So here, comes the use of "from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer".

It calculates the compound score of each and every tweet in the dataset. After calculating the compound score, if the score is above or equal to 0 it labels the tweet as 0 meaning "Non Cyberbullying Case" and for the tweets with score below 0 are labled as 1 meaning "Cyberbullying Case".

DATA SPLITTING

```
X=data['Summary_Clean']
Y=data['comp_score']
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.30,random_state=50)
```

Let's initialize the dataset and segregate into Training and Test set. Train set contains 70% of dataset and Test set contains the remaining 30% of the dataset.

FEATURE EXTRACTION

```
vector = CountVectorizer(stop_words = 'english', lowercase = True)
#fitting the data
```

```
training_data = vector.fit_transform(X_train)
#tranform the test data
testing_data = vector.transform(X_test)
```

The model will train on (X_train, Y_train) and it will get validated on (X_test, Y_test).

CLASSIFICATION

```
#=== LINEAR REGRESSION ===
#initialize the model
clf = LinearRegression()
#fitting the model
clf.fit(training_data, Y_train)
#predict the model
predictions = clf.predict(testing_data)
#=== PERFROMANCE ANALYSIS ===
#calculate accuracy
Error_value=metrics.mean_absolute_error(Y_test,predictions)
Accuracy_Linear=100-Error_value
print("********************************************")
print()
print("--- PERFORMANCE ANALYSIS ---")
print()
print(" Accuracy for Linear regression :",Accuracy_Linear,'%')
print()
#=== Stochastic Gradient descent ===
#initialize the model
sgd = SGDClassifier(loss="hinge", penalty="l2", max_iter=5)
#fitting the model
sgd.fit(training_data, Y_train)
#predict the model
predictions_sgd=sgd.predict(testing_data)
#=== PERFROMANCE ANALYSIS ===
#calculate accuracy
```

```
Error_value_sgd=metrics.mean_absolute_error(Y_test,predictions_sgd)
Accuracy_sgd=100-Error_value_sgd

print()
print("--- PERFORMANCE ANALYSIS ---")
print()
print(" Accuracy for Stochastic gradient descent :",Accuracy_sgd,'%')
print()
print("********************************************")
print()
```

Here comes the use of “from sklearn.linear_model import LinearRegression” and “from sklearn.linear_model import SGDClassifier” for implementing the Linear Regression and Stochastic Gradient Descent models.

PREDICTION

```
print("--- PREDICTION ---")
print()
print(predictions_sgd.shape)
z=int(input("Enter the id for finding Cyberbullying Case or Non Cyberbullying Case "))
if predictions_sgd[z] == 1:
    print('***********************************')
    print()
    print('--Cyberbullying cases --')
    print()
    # print('***********************************')
else:
    # print('***********************************')
    print()
    print('-- Non cyberbullying cases --')
    print()
    print('***********************************')
```

Here we are giving the index of the tweet in the test dataset to the model to predict wheter it is Cyberbullying Case or Non Cyberbullying Case.

VISUALIZATION

```
data['comp_score'] = data['comp_score'].replace([1 , 0] , ["Cyberbullying cases" , "Non Cyberbullying cases"])
plt.figure(figsize = (6,6))
counts = data['comp_score'].value_counts()
plt.pie(counts, labels = counts.index, startangle = 90, counterclock = False, wedgeprops = {'width' : 0.6},autopct='%1.1f%%', pctdistance = 0.55, textprops = {'color': 'black', 'fontsize' : 9}, shadow = True,colors = sns.color_palette("Paired")[3:])
plt.text(x = -0.35, y = 0, s = 'Total tweets: {}'.format(data.shape[0]))
plt.title('Analysing cyberbullying cases', fontsize = 14);
plt.show()
```

CHAPTER 7
OUTPUT SCREENSHOTS

OUTPUT

```
*********************************************

Data Selection

********************************************
   Unnamed: 0                        annotation  \
0           0  {'notes': '', 'label': ['1']}
1           1  {'notes': '', 'label': ['1']}
2           2  {'notes': '', 'label': ['1']}
3           3  {'notes': '', 'label': ['1']}
4           4  {'notes': '', 'label': ['1']}
5           5  {'notes': '', 'label': ['1']}
6           6  {'notes': '', 'label': ['1']}
7           7  {'notes': '', 'label': ['1']}
8           8  {'notes': '', 'label': ['1']}
9           9  {'notes': '', 'label': ['1']}

                                             content             user
0                             Get fucking real dude.    scotthamilton
1   She is as dirty as they come  and that crook ...         mattycus
2   why did you fuck it up. I could do it all day...          ElleCTF
3   Dude they dont finish enclosing the fucking s...           Karoli
4   WTF are you talking about Men? No men thats n...         joy_wolf
5  Ill save you the trouble sister. Here comes a ...          mybirch
6   Im dead serious.Real athletes never cheat don...             coZZ
7  ...go absolutely insane.hate to be the bearer ...  2Hood4Hollywood
8  Lmao  im watching the same thing ahaha. The ga...          mimismo
9  LOL  no he said  What do you call a jail cell ...   erinx3leannexo

********************************************
```

FIGURE 7.1: DATA SELECTION

```
********************************************

 Handling missing values

Unnamed: 0    0
annotation    0
content       0
user          0
dtype: int64

********************************************
```

FIGURE 7.2: DATA PREPROCESSING

```
********************************************

Before applying NLP techniques

0                                Get fucking real dude.
1      She is as dirty as they come  and that crook ...
2      why did you fuck it up. I could do it all day...
3      Dude they dont finish enclosing the fucking s...
4      WTF are you talking about Men? No men thats n...
5     Ill save you the trouble sister. Here comes a ...
6      Im dead serious.Real athletes never cheat don...
7     ...go absolutely insane.hate to be the bearer ...
8     Lmao  im watching the same thing ahaha. The ga...
9     LOL  no he said  What do you call a jail cell ...
Name: content, dtype: object

********************************************
```

FIGURE 7.3: BEFORE NLP

```
********************************************

After applying NLP techniques

0                                get fucking real dude
1    she is as dirty as they come and that crook re...
2    why did you fuck it up i could do it all day t...
3    dude they dont finish enclosing the fucking sh...
4    wtf are you talking about men no men thats not...
5    ill save you the trouble sister here comes a b...
6    im dead serious real athletes never cheat don ...
7    go absolutely insane hate to be the bearer of ...
8    lmao im watching the same thing ahaha the gay ...
9    lol no he said what do you call a jail cell to...
Name: Summary_Clean, dtype: object

********************************************
```

FIGURE 7.4: AFTER NLP

```
********************************************

--- PERFORMANCE ANALYSIS ---

 Accuracy for Linear regression : 99.61573301278705 %


--- PERFORMANCE ANALYSIS ---

 Accuracy for Stochastic gradient descent : 99.90634894184302 %

********************************************
```

FIGURE 7.5: PERFORMANCE ANALYSIS

```
********************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case
```

FIGURE 7.6: PREDICTION INPUT

```
********************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 1952
```

FIGURE 7.7: GIVEN INPUT

```
********************************************

--- PREDICTION ---

(6001,)
Enter the id for finding Cyberbullying Case or Non Cyberbullying Case 1952
***********************************

--Cyberbullying case --
```

FIGURE 7.8: PREDICTION OUTPUT

```
***********************************
```

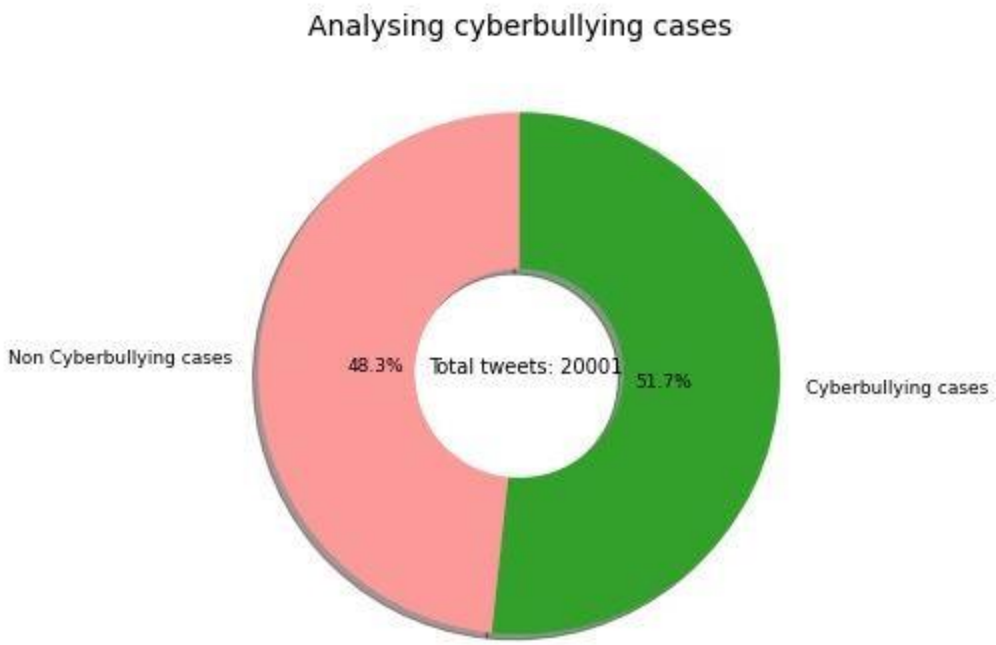

Figure 7.9: ANALYSIS

CHAPTER 8

CONCLUSION

We have developed an approach towards the detection of cyberbullying behaviour. If we are able to successfully detect such posts which are not suitable for adolescents or teenagers, we can very effectively deal with the crimes that are committed using these platforms. An approach is proposed for detecting and preventing Twitter cyberbullying using Supervised Binary classification Machine Learning algorithms. Our model is evaluated on both Linear Regression and Stochastic Gradient Descent, also for feature extraction, we used the Count vectorizer. Our model will help people from the attacks of social media bullies.

CHAPTER 9

FUTURE WORK

In future, it is possible to provide extensions or modifications to the proposed clustering and classification algorithms to achieve further increased performance. Apart from the experimented combination of data mining techniques, further combinations and other clustering algorithms can be used to improve the detection accuracy and to reduce the rate offensive tweets. Finally, the cyberbullying detection system can be extended as a prevention system to enhance the performance of the system.

REFERENCES

1. Amanpreet Singh, Maninder Kaur, "Content-based Cybercrime Detection: A Concise Review", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8, pages 1193-1207, 2019.
2. Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. "Detecting offensive language in social media to protect adolescent online safety". In Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), pages 71– 80. IEEE, 2012.
3. NaliniPriya. G and Asswini. M (2015) "A Dynamic Cognitive System For Automatic Detection And Prevention Of Cyber-Bullying Attacks", ARPN Journal of Engineering and Applied Sciences ©2006-2015 Asian Research Publishing Network (ARPN). VOL. 10, NO. 10, JUNE 2015.
4. "Protective shield shield for social networks to defend cyberbullying and online grooming attacks" in Proceedings of 40 th IRF International Conference, Pune, India, ISBN: 978-93-85832-16-1, 2015.
5. Gupta, P. Kumaraguru, and A. Sureka, "Characterizing pedophile conversations on the internet using online grooming," CoRR, vol. abs/1208.4324, 2012.
6. K. Jedrzejewski and M. Morzy, "Opinion Mining and Social Networks: A Promising Match," 2011 Int. Conf. Adv. Soc. Networks Anal. Min., pp. 599–604, Jul. 2011.
7. H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Analyzing Labeled Cyberbullying Incidents on the Instagram Social Network."
8. Kelly Reynolds, April Kontostathis, Lynne Edwards, "Using Machine Learning to Detect Cyberbullying", 2011 10th International Conference on Machine Learning and Applications volume 2, pages 241–244. IEEE, 2011.
9. Edosomwan, S.; Prakasan, S.K.; Kouame, D.; Watson, J.; Seymour, T. The history of social media and its impact on business. J. Appl. Manag. Entrep. 2011, 16, 79–91.
10. Bauman, S. Cyberbullying: What Counselors Need to Know; John Wiley & Sons: Hoboken, NJ, USA, 2014.
11. Pereira-Kohatsu, J.C.; Quijano-Sánchez, L.; Liberatore, F.; Camacho-Collados, M. Detecting and Monitoring Hate Speech in Twitter. Sensors 2019, 19, 4654.
12. Miller, K. Cyberbullying and its consequences: How cyberbullying is contorting the minds of victims and bullies alike, and the law's limited available redress. S. Cal. Interdisc. Law J. 2016, 26, 379.

13. Price, M.; Dalgleish, J. Cyberbullying: Experiences, impacts and coping strategies as described by australian young people. Youth Stud. Aust. 2010, 29, 51.
14. Smith, P.K. Cyberbullying and Cyber Aggression. In Handbook of School Violence and School Safety; Informa UK Limited: Colchester, UK, 2015.
15. Sampasa-Kanyinga, H.; Roumeliotis, P.; Xu, H. Associations between Cyberbullying and School Bullying Victimization and Suicidal Ideation, Plans and Attempts among Canadian Schoolchildren. PLoS ONE 2014, 9, e102145.
16. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. arXiv 2017, arXiv:1703.04009. 9. Mc Guckin, C.; Corcoran, L. (Eds.) Cyberbullying: Where Are We Now? A Cross-National Understanding; MDPI: Wuhan, China, 2017
17. Vaillancourt, T.; Faris, R.; Mishna, F. Cyberbullying in Children and Youth: Implications for Health and Clinical Practice. Can. J. Psychiatry 2016, 62, 368–373. [CrossRef]
18. Görzig, A.; Ólafsson, K. What Makes a Bully a Cyberbully? Unravelling the Characteristics of Cyberbullies across Twenty-Five European Countries. J. Child. Media 2013, 7, 9–27. [CrossRef]
19. Salton, G.; Buckley, C. Term-wsevening approaches in automatic text retrieval. Inf. Process. Manag. 1988, 24, 513–523.
20. Liu, Q.; Wang, J.; Zhang, D.; Yang, Y.; Wang, N. Text Features Extraction based on TF-IDF Associating Semantic. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 2338–2343.
21. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
22. Goldberg, Y.; Levy, O. word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv 2014, arXiv:1402.3722.
23. Li, J.; Huang, G.; Fan, C.; Sun, Z.; Zhu, H. Key word extraction for short text via word2vec, doc2vec, and textrank. Turk. J. Electr. Eng. Comput. Sci. 2019, 27, 1794–1805. [CrossRef]
24. . Jiang, C.; Zhang, H.; Ren, Y.; Han, Z.; Chen, K.-C.; Hanzo, L. Machine Learning Paradigms for Next-Generation Wireless Networks. IEEE Wirel. Commun. 2016, 24, 98–105. [CrossRef]
25. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. Ain Shams Eng. J. 2014, 5, 1093–1113. [CrossRef]

26. Al-Garadi, M.A.; Hussain, M.R.; Khan, N.; Murtaza, G.; Nweke, H.F.; Ali, I.; Mujtaba, G.; Chiroma, H.; Khattak, H.A.; Gani, A. Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges. IEEE Access 2019, 7, 70701–70718. [CrossRef]
27. Maalouf, M. Logistic regression in data analysis: An overview. Int. J. Data Anal. Tech. Strat. 2011, 3, 281–299.