**Predicting Cyberbullying on Social Media in the**

**Big Data Era Using Machine**

**Abstract:**

Cyberbullying is the use of Information and Communication Technology (ICT) by individuals to humiliate, tease, embarrass, taunt, defame and disparage a target without any face-to-face contact. Social media is the “virtual playground” used by bullies with the upsurge of social networking sites such as Facebook, Instagram, YouTube, Twitter etc. It is critical to implement models and systems for automatic detection and resolution of bullying content available online as the ramifications can lead to a societal epidemic. This research proffers a novel hybrid model for Cyberbullying detection in three different modalities of social data, namely, textual, and info-graphic (text embedded along with an image). The architecture consists of a Deep Learning convolution neural network (DLCNN) for predicting the textual bullying content. The info-graphic content is discretized by separating text from the image using Google Lens of Google Photos App. The processing of textual and visual components is carried out using the hybrid architecture and a Boolean system with a logical OR operation is augmented to the architecture which validates and categorizes the output on the basis of text and image bullying truth value. The model achieves a prediction accuracy of 98% which is acquired after performing tuning of different hyper-parameters. The simulation results show that the proposed method gives the better accuracy compared to the state of art approaches.

**Introduction**

Through Cyberbullying, an individual or victim can be humiliated or hurt before whole network on the web[1]. It bothers the psychological and physical condition of an individual because of which an expansive number of suicides and discouragement cases happen. These days, Cyberbullying, through images or memes are extremely common. Pornographic images or pictures with oppressive, mean or defamatory remarks are being presented on one's profile in order to bully them. Protective measures must be taken so as to control this. Thus, in our work we have made an automated framework which will, in addition to detection of bullying for text [2], also recognizes the bullying in the pictures. There can be a straightforward picture with no content on it or there might be such sort of picture also where some content is embedded with that picture. More recently, as memes and GIFs [3] dominate the social feeds; typo-graphic and info-graphic visual content has become a considerable element of social data. Thus, cyber bullying, through varied content modalities is very common. Researchers worldwide have been trying to develop new ways to detect cyber bullying, manage it and reduce its prevalence on social media. Advanced analytical methods and computational models for efficient processing, analysis and modeling for detecting such bitter, taunting, abusive or negative content in images, memes or text messages are imperative. Social media specificity, topic dependence and variety in hand-crafted features currently define the bottlenecks in detecting online bullying posts [4]. State-of-the-art results are achieved by deep learning methods on some specific language problems by using the capabilities of hierarchical learning and generalization [5]. Pertinent studies report the use of deep learning models like CNN[6], RNN[]7 and semantic image features for bullying content detection by analyzing textual, image based and user features [8].But most of the research on online cyber-aggression, harassment detection and toxicity has been limited to text-based analytics [9]. Visual analysis of images have also been reported by few related studies for determining bullying

content [10-12] but the domain of visual text which combines both text and image has been least explored in literature. The combination can be observed in two variants: typo-graphic (artistic way of text representation) or info-graphic (text embedded along with an image). This research work puts forward a hybrid deep learning model for bullying content prediction, where the content, c ɛ {text, image, info-graphic}.The primary contribution of the work is that unlike previous models which are mono-modal dealing with a single type of content modality.

The major contributions of the paper as follows:

* The proposed method is capable of detecting the bullying data from the both Textual and info-graphic images respectively.
* The overall purpose of this analysis is therefore to evaluate the performance of standard DLCNN classification algorithms by using SIFT features for bullying language identification.

This paper is summarized follows through as: In Section 2, literature review for cloud data security with the comparison of methodology with defining problem, implication, merits and demerits. Section 3 gives the detailed information about the proposed methodology. Section 4 discusses about the results analysis and finally Section 5, concluded the summarization of whole paper.

**Literature Survey**

Twitter is an online social networking platform, also referred to as a microblogging site, where users can share information via messages up to 140 characters. These short messages are called „tweets‟. Twitter allows users to tweet through its website or through its applications developed for various external compatible devices. In most countries, users can also use SMS services to tweet. Tweets can be read by anyone unless the users restrict access strictly to their followers. When a user subscribes to another user account, the subscription is termed „following‟ and the subscriber is called the „follower‟. Twitter, as a social networking platform, spins around the term „followers‟. For example, if user A is following user B, user A as a follower gets access to read and retweet user B‟s tweets. Out of all the tweets generated on Twitter, roughly 40% of tweets are conversational tweets (Kelly, 2009). Users make use of hashtags to tweet about trending topical information. Similarly, users make use of „@‟ followed by a username, for example „@username‟ to post a tweet mention or reply to another user.

One thing that makes it easy for bullies to harass someone online is that they can retain their anonymity by creating fake accounts to bully someone. Due to the functionality of „hashtags‟ and „@username‟, victims are more vulnerable to direct online attacks. In addition, the victims are totally exposed, as their followers can witness the entire cyberbullying episodes. Twitter provides a system to reduce cyberbullying, but unfortunately it is not effective. Twitter has a „report abuse‟ form that users affected by cyberbullying must fill out if they wish some action needs to be taken. Twitter needs a more intelligent system to detect cyberbullying, which is more efficient in detecting cyberbullying tweets on Twitter.

As the negative influence caused by cyberbullying is increasing, an increasing number of studies are dedicated to dealing with, mainly detection of, cyberbullying. There are nontechnical works concentrating on giving definitions, reporting status quo and understanding the problem of cyberbullying [13]. Those studies give directions for our detection work. There are still not many studies dedicated to automatically detect cyberbullying but the number is increasing [14].The causes of cyberbullying and its prevalence, especially for children and young adults, have been extensively studied in social sciences research [15]. In terms of its impact, empirical studies have demonstrated a link between suicidal ideation among adolescents and experiences with cyberbullying [16]. The characteristic profiles of offenders and victims in cyberbullying are presented in [17]. This paper also discusses the possible strategies of prevention and intervention. Those studies enlighten us on the scope and spread the awareness of the problem.

The research introduced above all focuses on the detection of bullies, but in this work, our target is different. As mentioned before, we attempt to predict if an image posted on social media would become a target for the bullies. Therefore, instead of spotting the bullies, we focus on preventing people from feeding the bullies. Moreover, instead of text information, we utilize image features to build classification model. The expected outcome is a model that can warn social network site users before they post any images that are highly “bullyable”. However, as far as we know, there is not any other work that is based on image features in cyberbullying detection. In this work, we explore the potential of existing image feature extracting methods in other area and make them function in our work.

**Existing System:**

There exists various child predator detection system which are used in gaming, audio chat and in various online entertainment platform. While playing games or for using online audio chat there exists a child predator system which detects an online sexual harassment and prevent child from getting abused or getting harassed by sexual predator as this existing system is only used when the children are playing games on internet or doing any audio chats. As now we are in internet era various children are now days using social media platform for various social activities. They are mostly active on social media so to prevent child harassment we need a child predator detection system for social media.

In existing system use 5 classification algorithm Neural Network classifier on the TF-IDF feature set, or the conversation-based approach using the Ridge or the Naıve Bayes classifier on the TF-IDF feature set.

In our system, we will implement only one algorithm for image and text classification. We will give more accuracy as compare to existing system because of The Support Vector Machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems.

**Proposed System:**

In this project we are using various machine learning algorithms such as SVM, Random Forest, Naïve Bayes, KNearest Neighbours, and Decision Tree to predict child harasser’s posts from social networks.

Using all algorithms we will build train model with normal and harasser’s word and messages and this train model will applied on new posts from users to predict whether new post is normal or contain harasser’s stuff.

**3.3 SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : MINIMUM i3.
* Hard Disk : 40 GB.
* Ram : 4 GB.

**SOFTWARE REQUIREMENTS:**

* **Operating System:** Windows 8
* **Coding Language**: Python 3.7

**3.4 SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**4.SYSTEM DESIGN**

**4.1 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

User



Admin



**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**COLLABORATION DIAGRAM:**



**4.3 IMPLEMENTATION:**

**Modules:**

User module:

using this module users can create an account. Using account details they can login to application and then send and view posts.

Admin Module:

Admin can view all registered user account and then accept or reject new user account. Admin responsible to add new harasser/non-harasser messages to machine learning train dataset. Admin has to run all or at-least one SVM algorithm to perform harasser’s message detection from user side. Admin can view or monitor all posts send by all users.

**5.SOFTWARE ENVIRONMENT**

# What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/" \t "_blank)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

**Advantages of Python :-**

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you**[download Python](https://data-flair.training/blogs/install-python-windows/)** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### **Advantages of Python Over Other Languages**

#### 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

#### 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **[machine learning](https://data-flair.training/blogs/machine-learning-tutorials-home/)**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/" \t "_blank), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/" \t "_blank)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/" \t "_blank), [Scikit-learn](https://scikit-learn.org/stable/" \t "_blank), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as **[Fork Python](https://practice.geeksforgeeks.org/courses/fork-python" \t "_blank)** available Free on GeeksforGeeks.

### **Step 2 – Learn Various ML Concepts**

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### **Advantages of Machine learning :-**

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

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. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As **[ML algorithms](https://data-flair.training/blogs/machine-learning-algorithms/)** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### **Disadvantages of Machine Learning :-**

#### 1. Data Acquisition

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.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

**[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning)** is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software" \o "Free software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks" \o "Neural networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google" \o "Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain" \o "Google Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License" \o "Apache License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license" \o "Open-source license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **[https://www.python.org](https://www.python.org/)**



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**6.SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

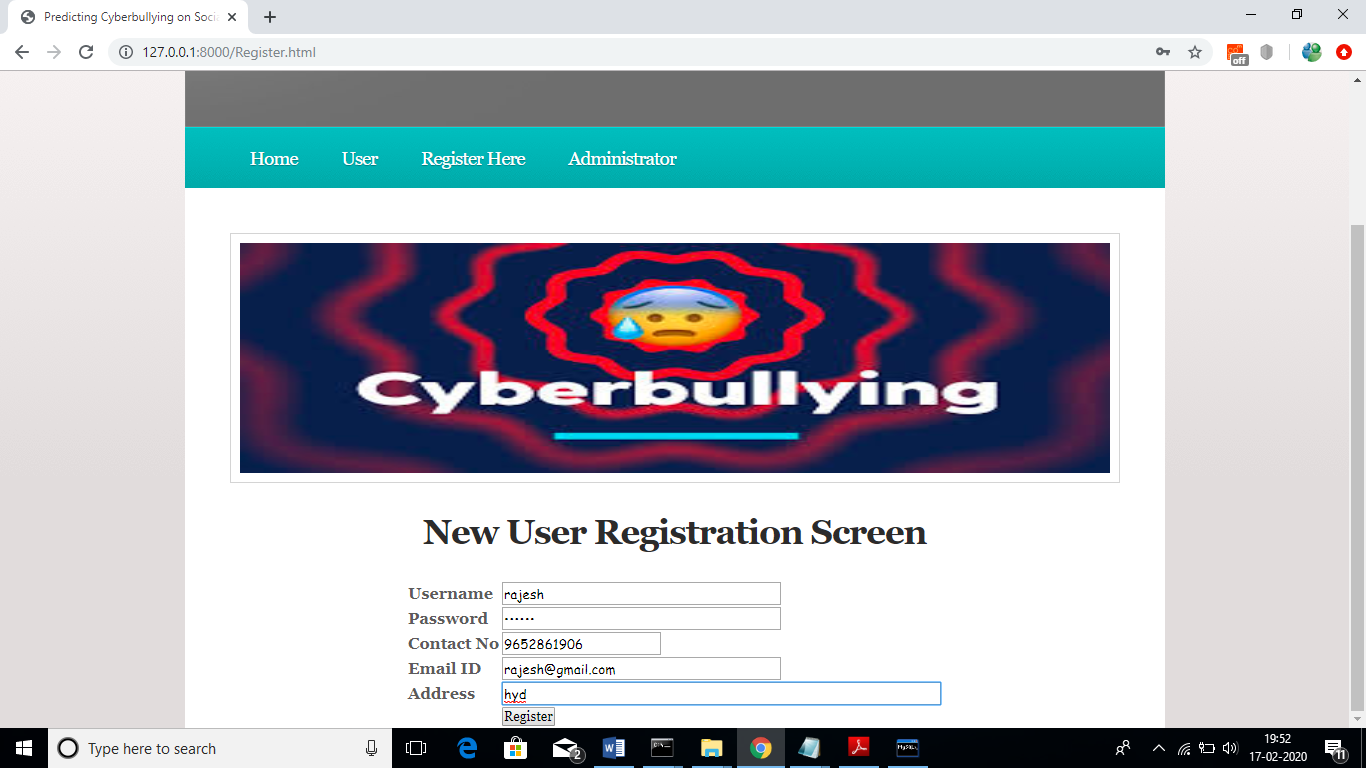
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

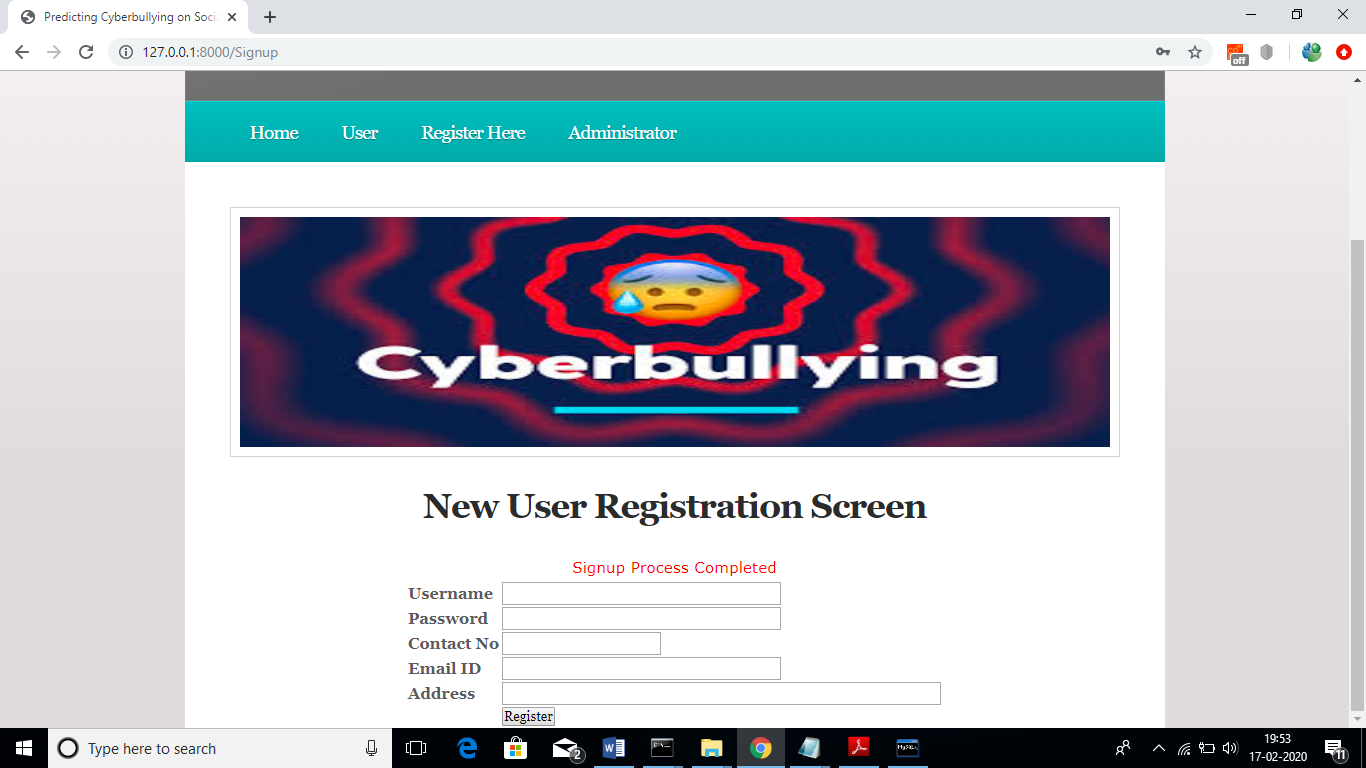
**7.SCREENSHOTS**



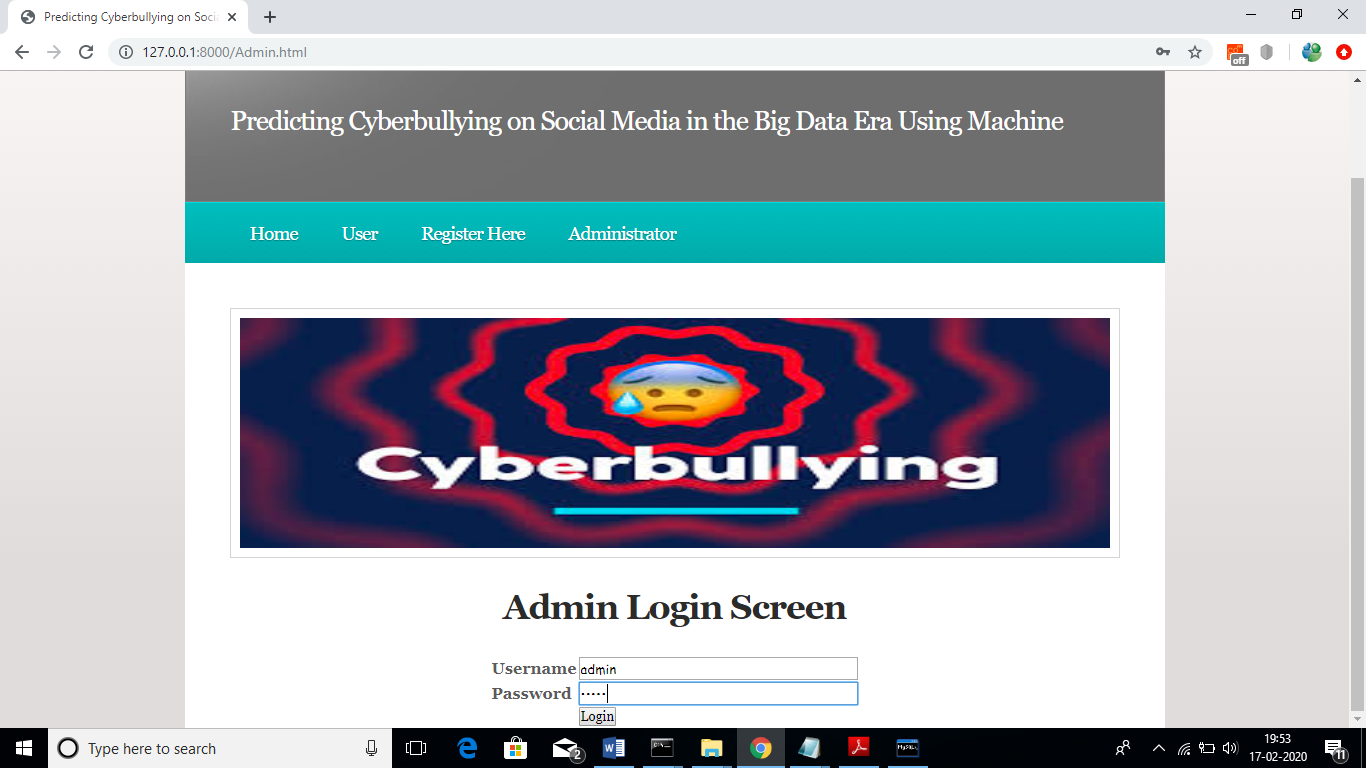
In above screen click on ‘Register Here’ link and add new user to create account



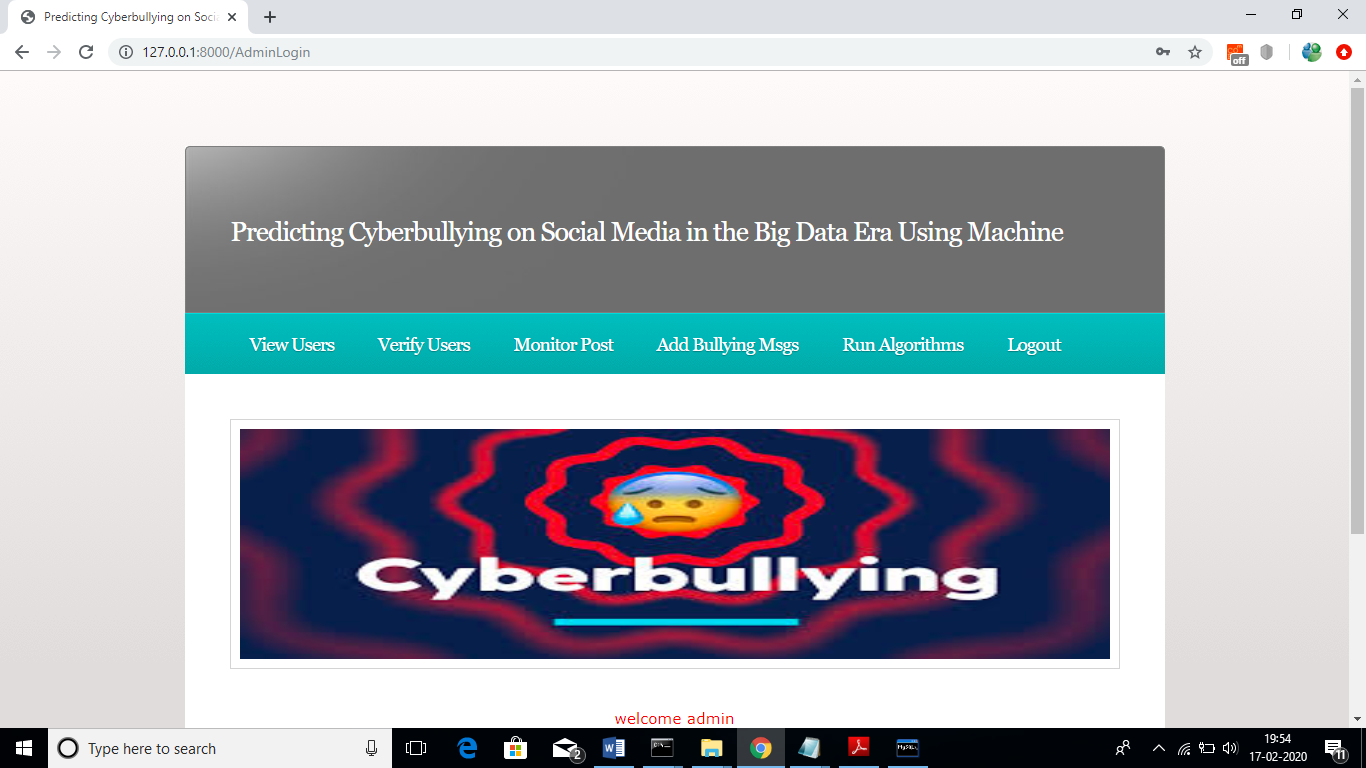
In above screen now click on ‘Register’ button to add details



In above screen sign up process completed. Now click on ‘Administrator’ link to login as admin and give permission to new user



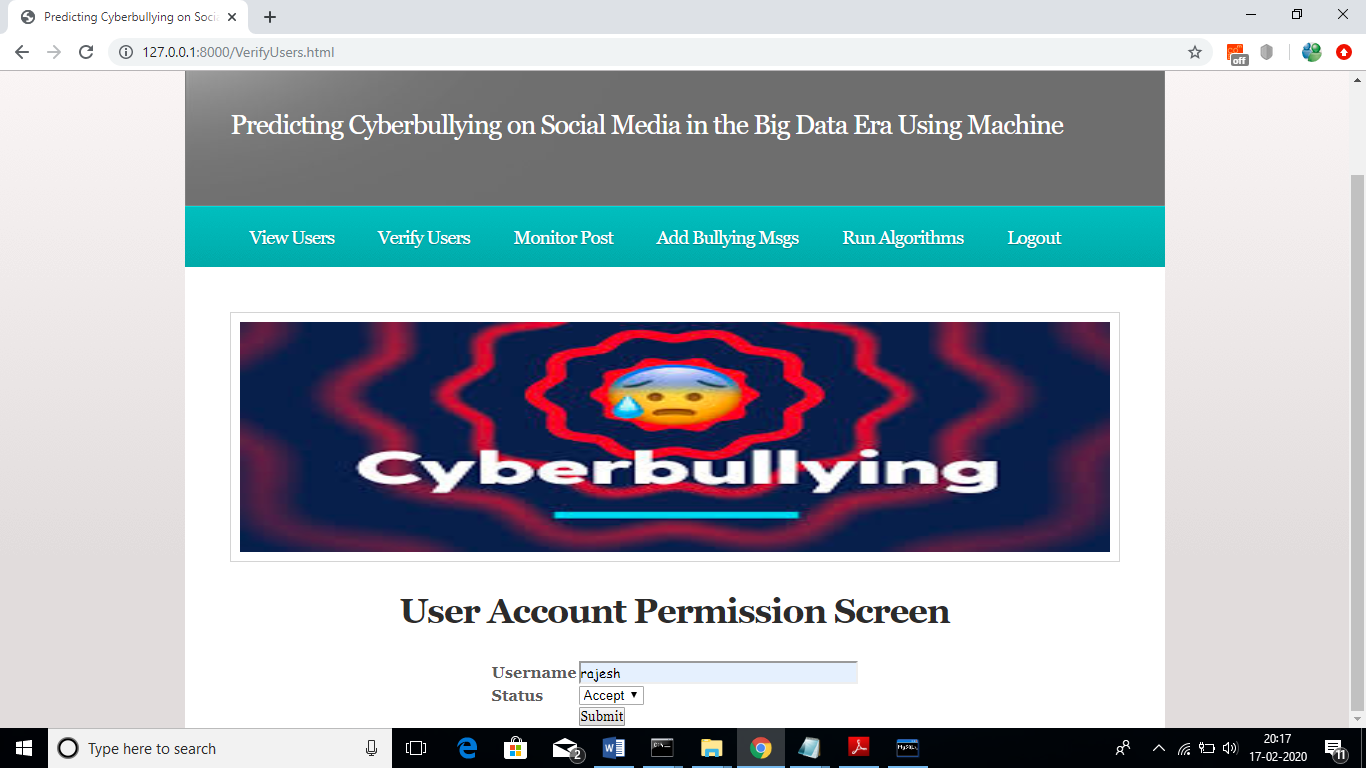
In above screen login as ‘admin’ by giving username as ‘admin’ and password as ‘admin’. After login will get below screen



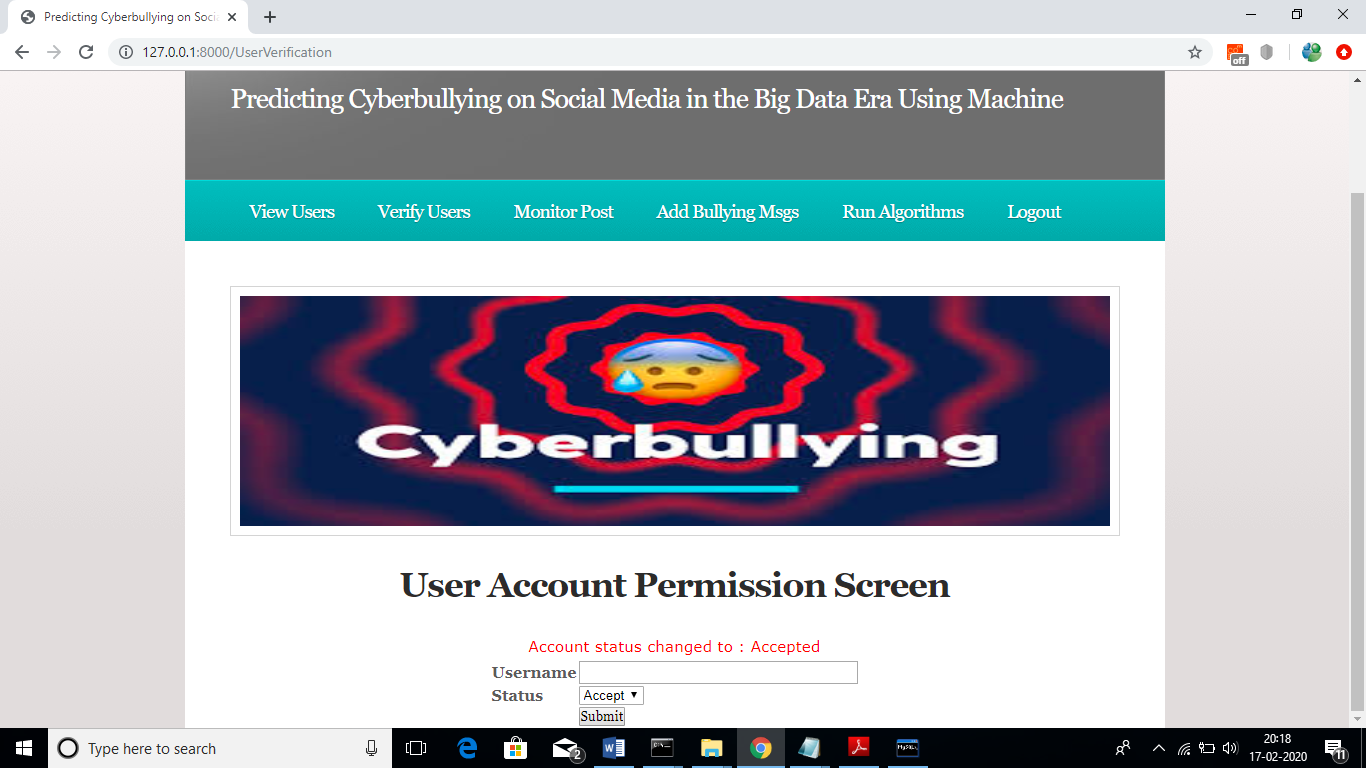
Now admin can click on ‘View Users’ link to view all users list



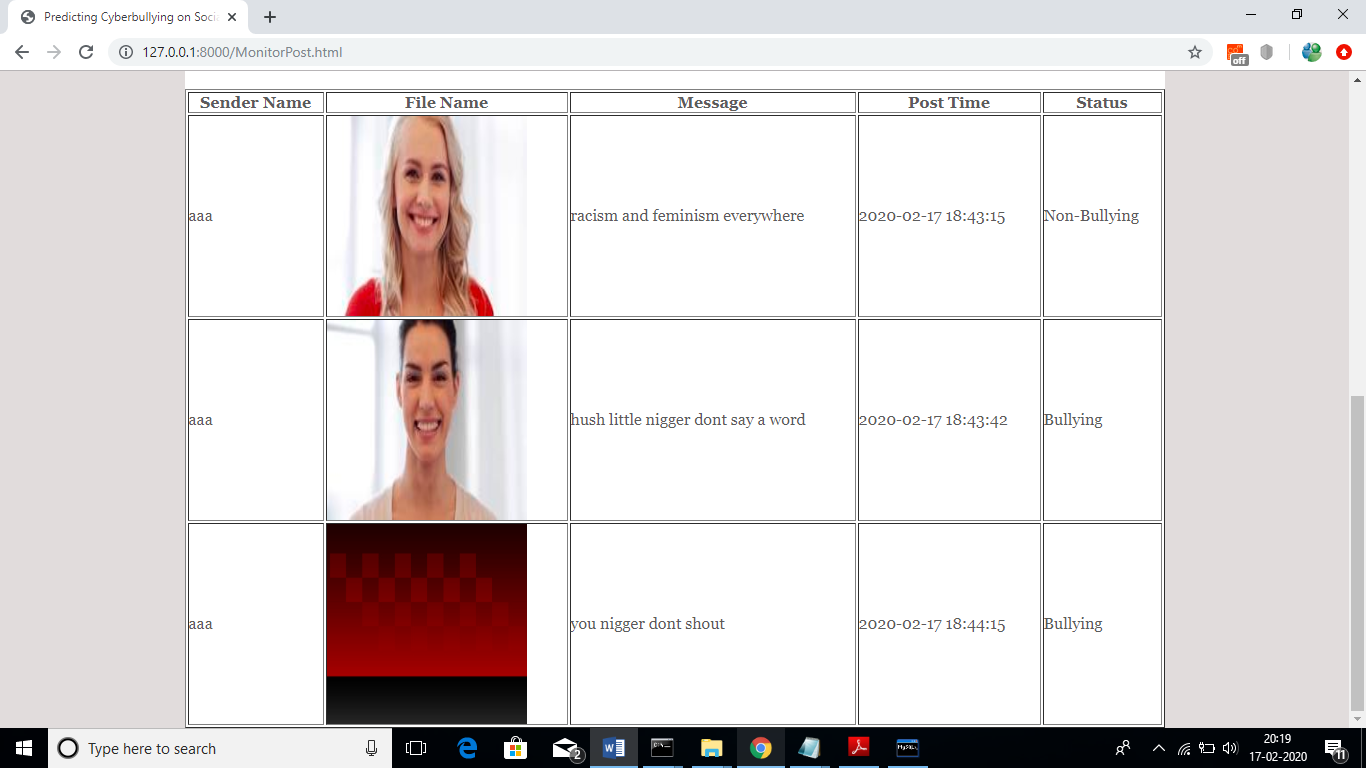
In above screen we can see ‘rajesh’ account is in pending state and to give permission to rajesh. Now admin will click on ‘Verify Users’ link to get below screen and to give permission



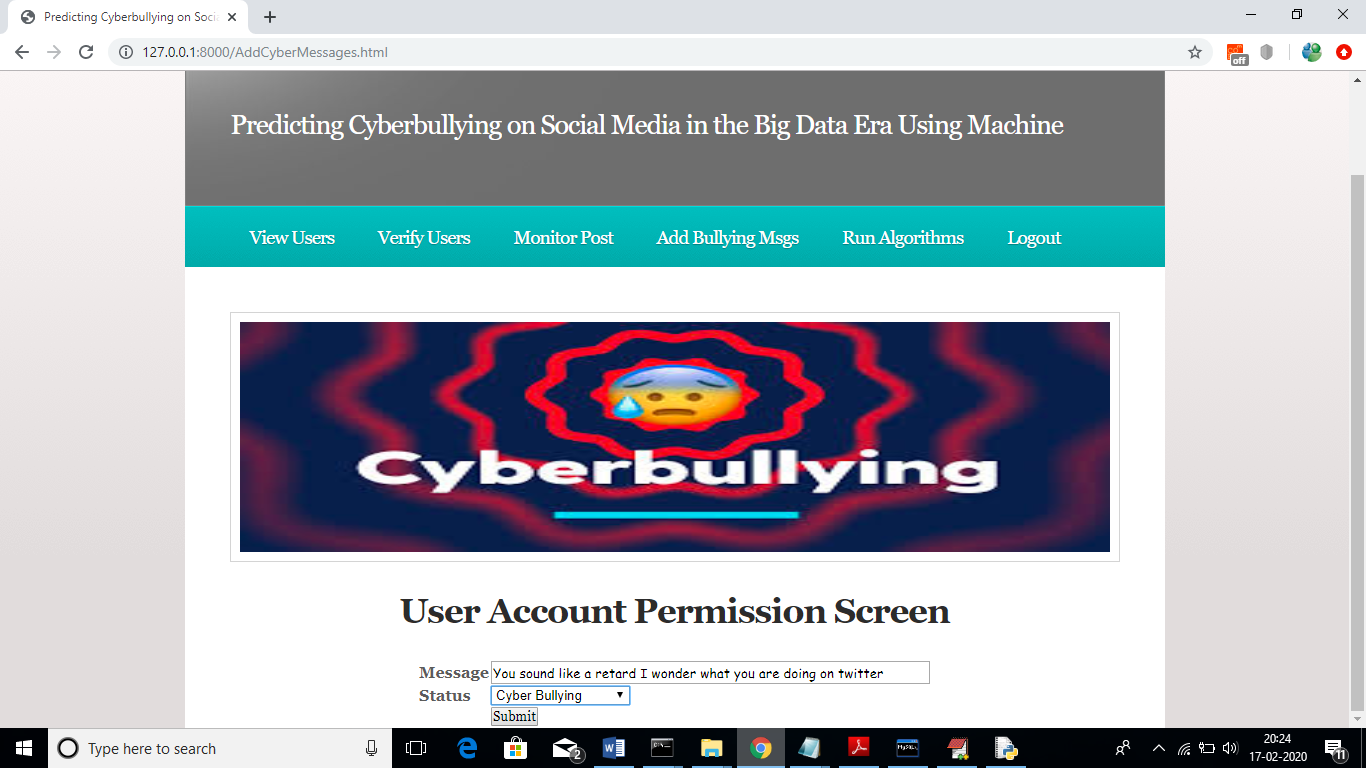
In above screen admin will enter username and then select ‘Accept’ or ‘Reject’ option to give permission.



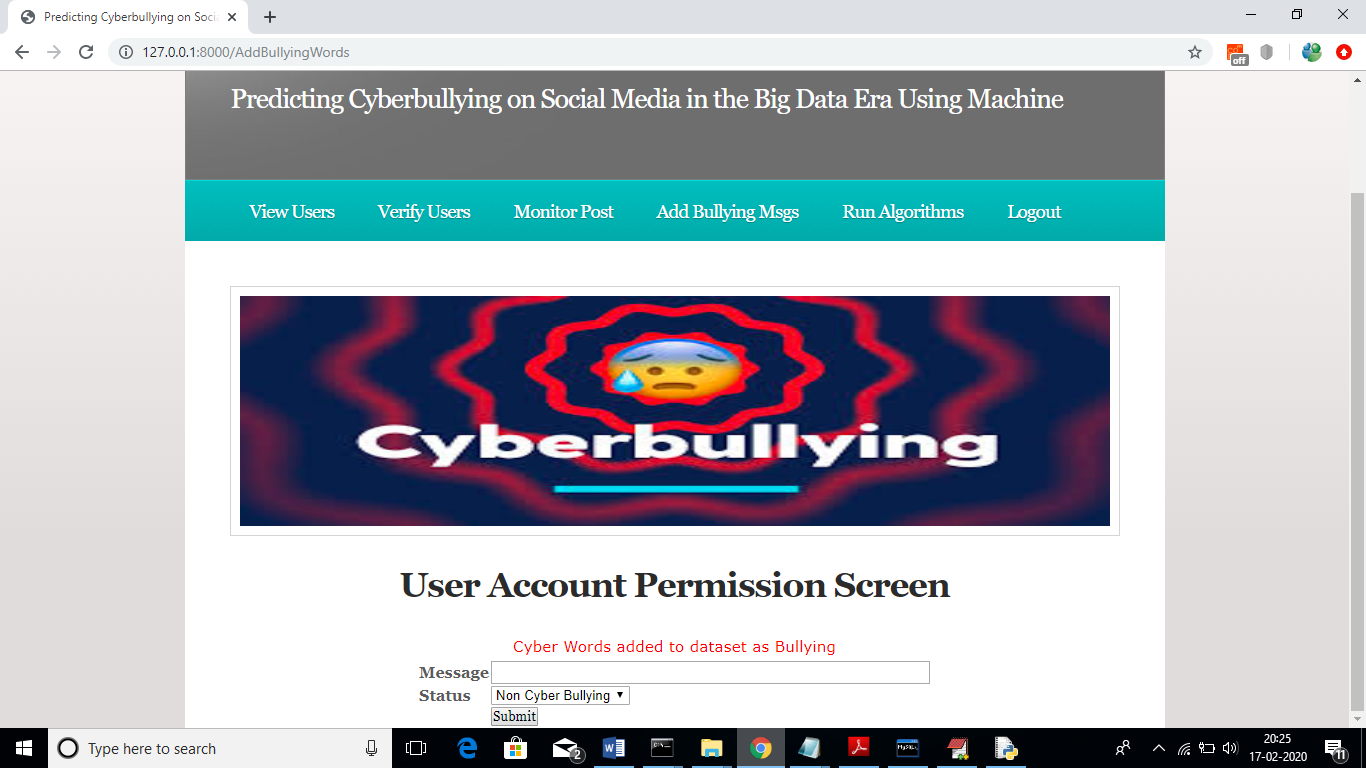
In above screen account state changed to ‘Accepted’. Now admin can click on ‘Monitor Post’ to view all post from past users



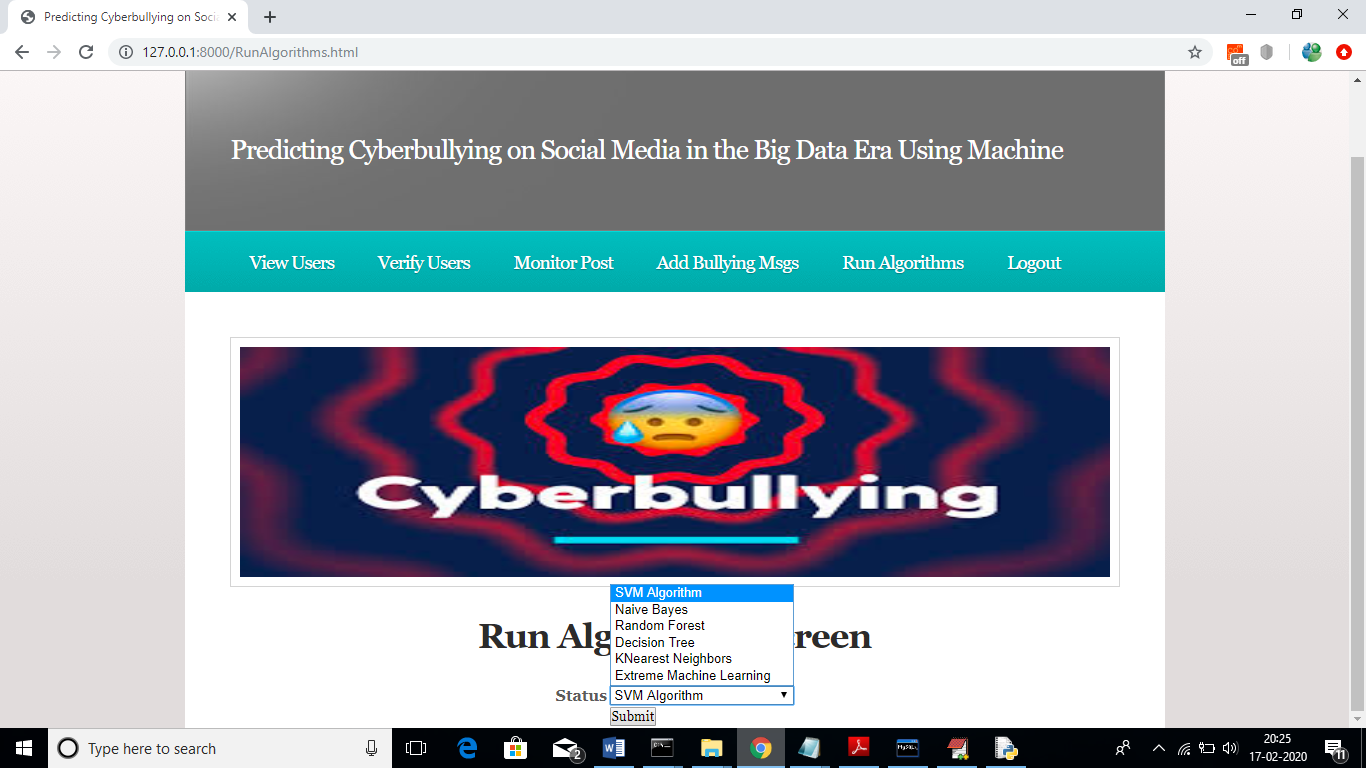
In above screen application will automatically detect whether message is non-bullying or bullying from machine learning algorithms. Now admin can click on ‘Add Bullying Msgs’ link to add words



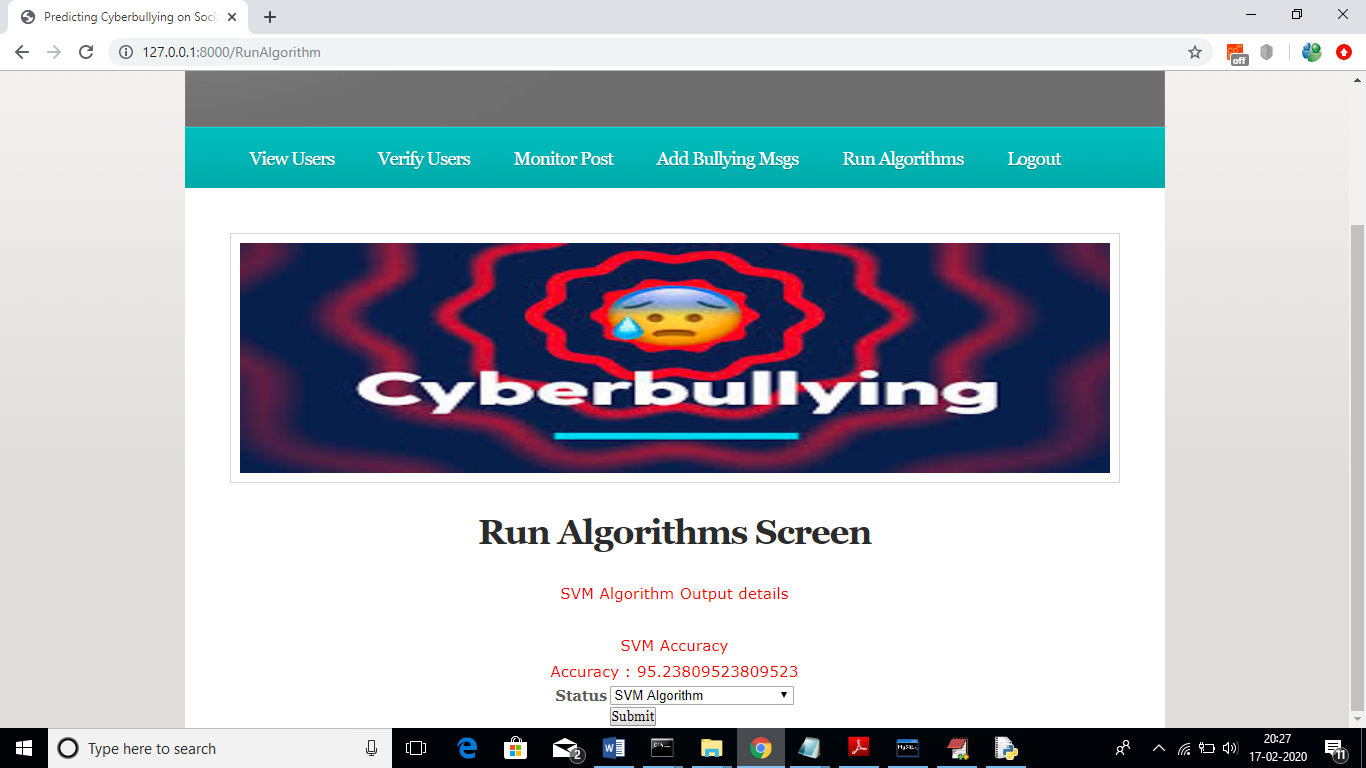
In above screen admin adding one sentence as ‘Cyber Bullying’ and similarly he can add all possible bullying and non-bullying messages. After adding messages will get below screen



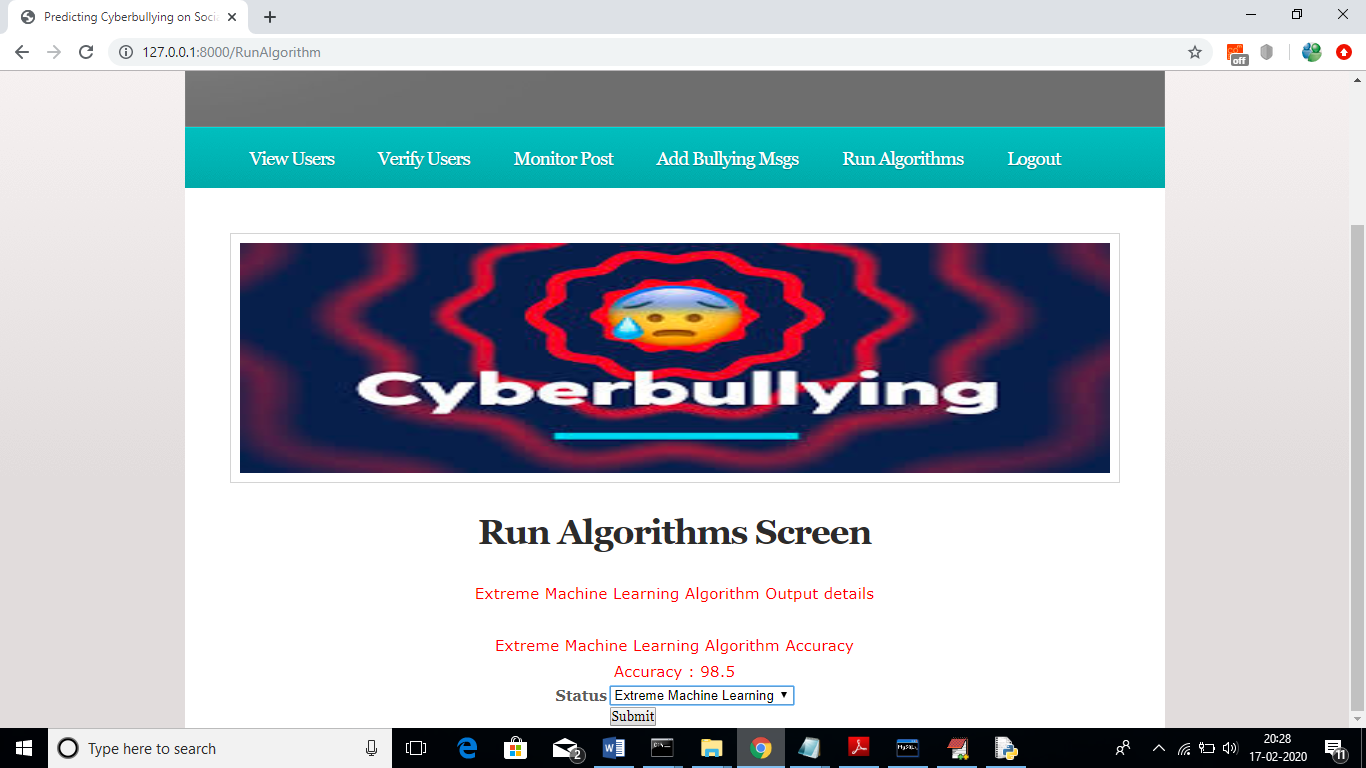
Now admin can click on ‘Run Algorithms’ link to generate train model using entire dataset to predict user posts as normal or bullying



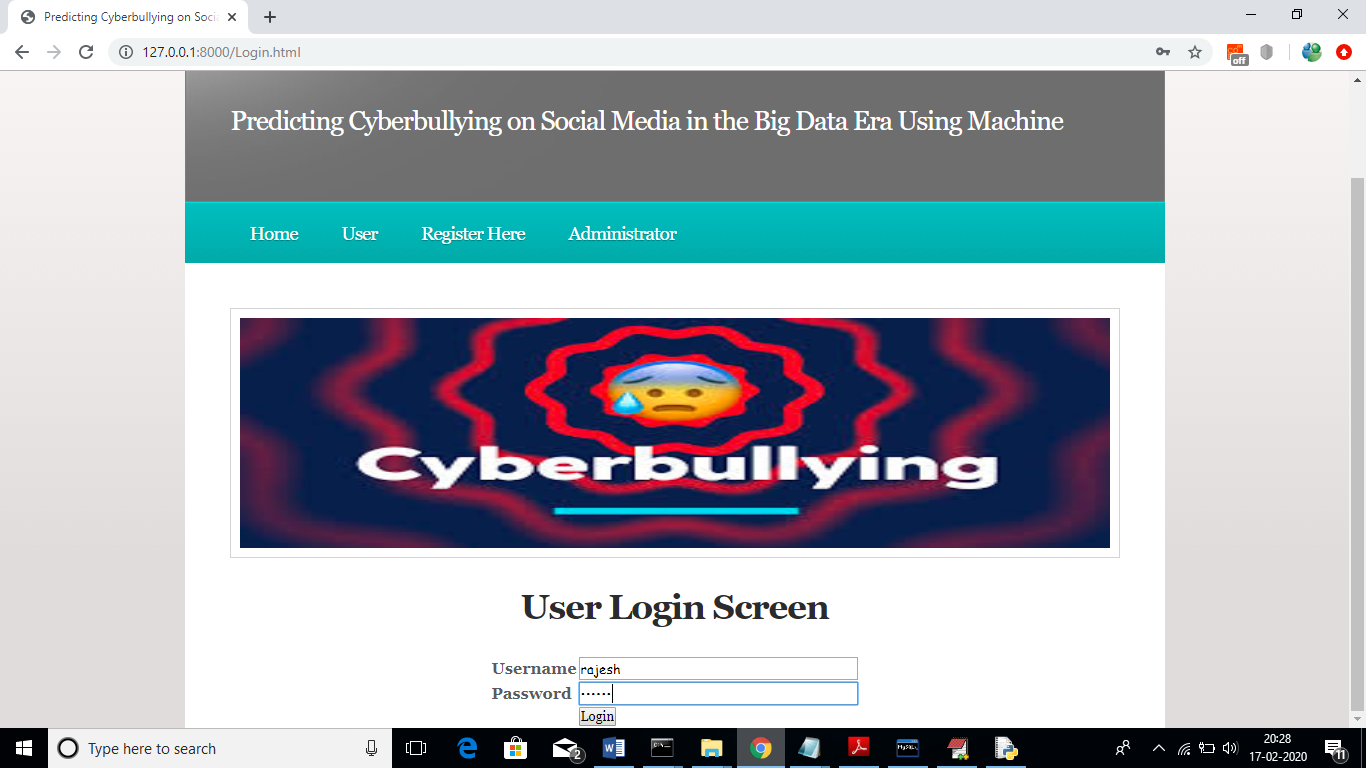
In above screen admin has to select each algorithm and click on ‘Submit’ button to train model and we will get accuracy also for each algorithm. Admin has to repeat this step whenever first time he starts the server or upon adding new bulling messages.



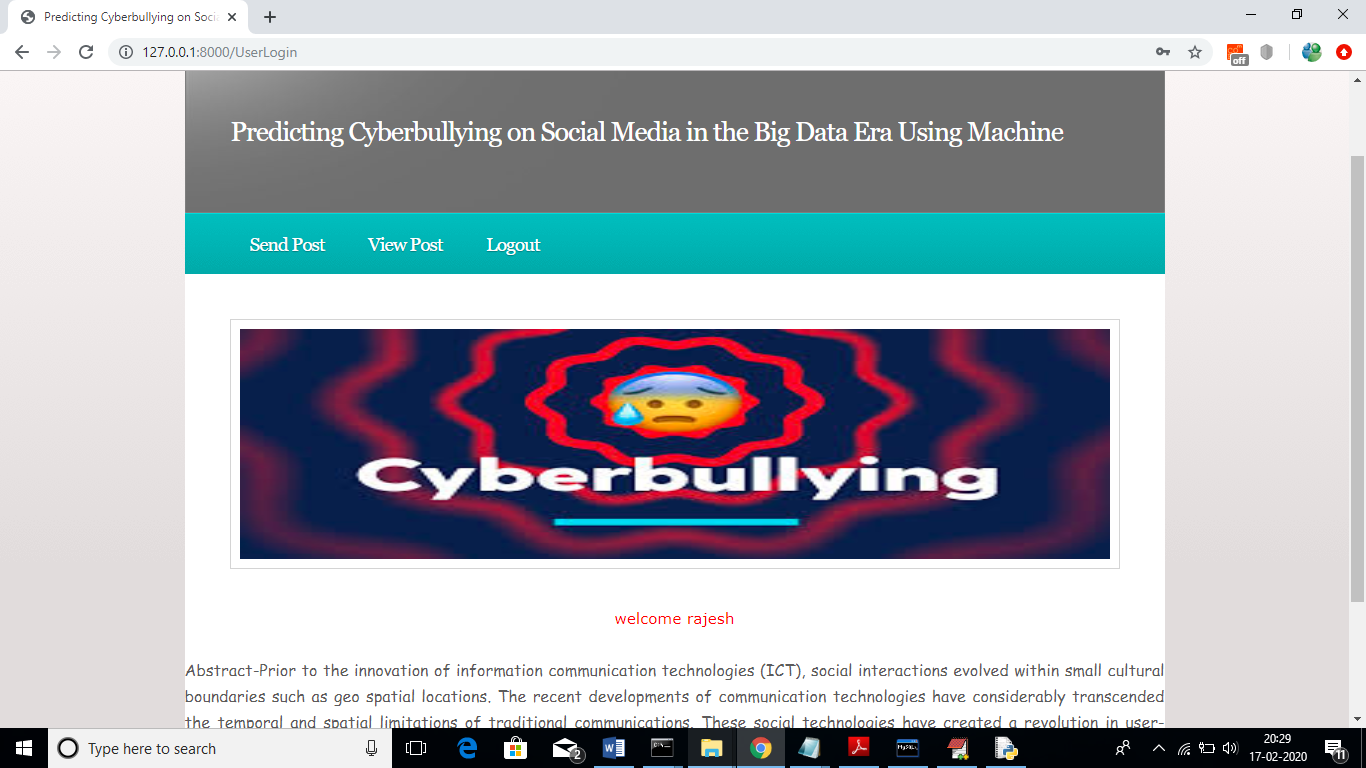
In above screen I ran SVM and got accuracy as 95. Similarly u need to select all algorithms one by one and run it.



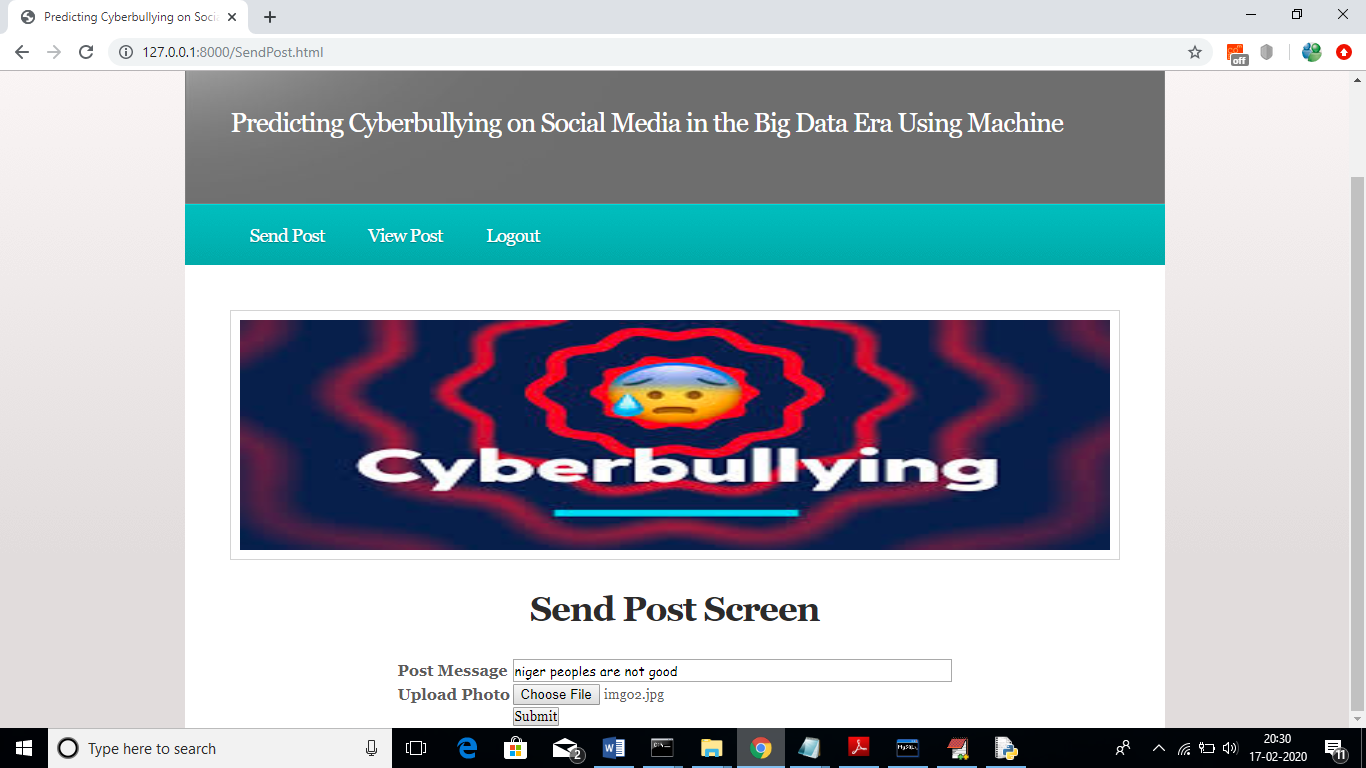
In above screen advance ‘Extreme Machine Learning’ algorithm gave 98% accuracy. Now admin logout and login as user to send posts.



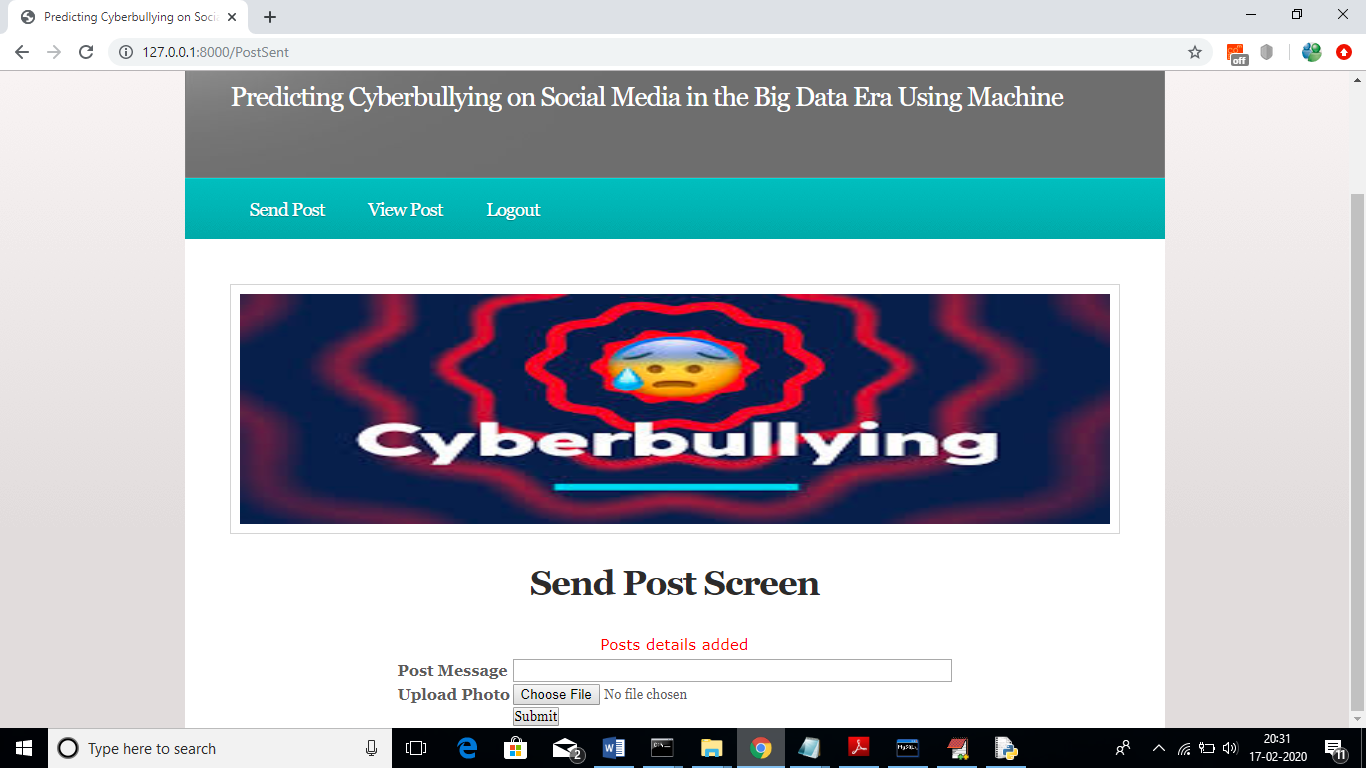
In above screen rajesh user is login and after login will get below screen

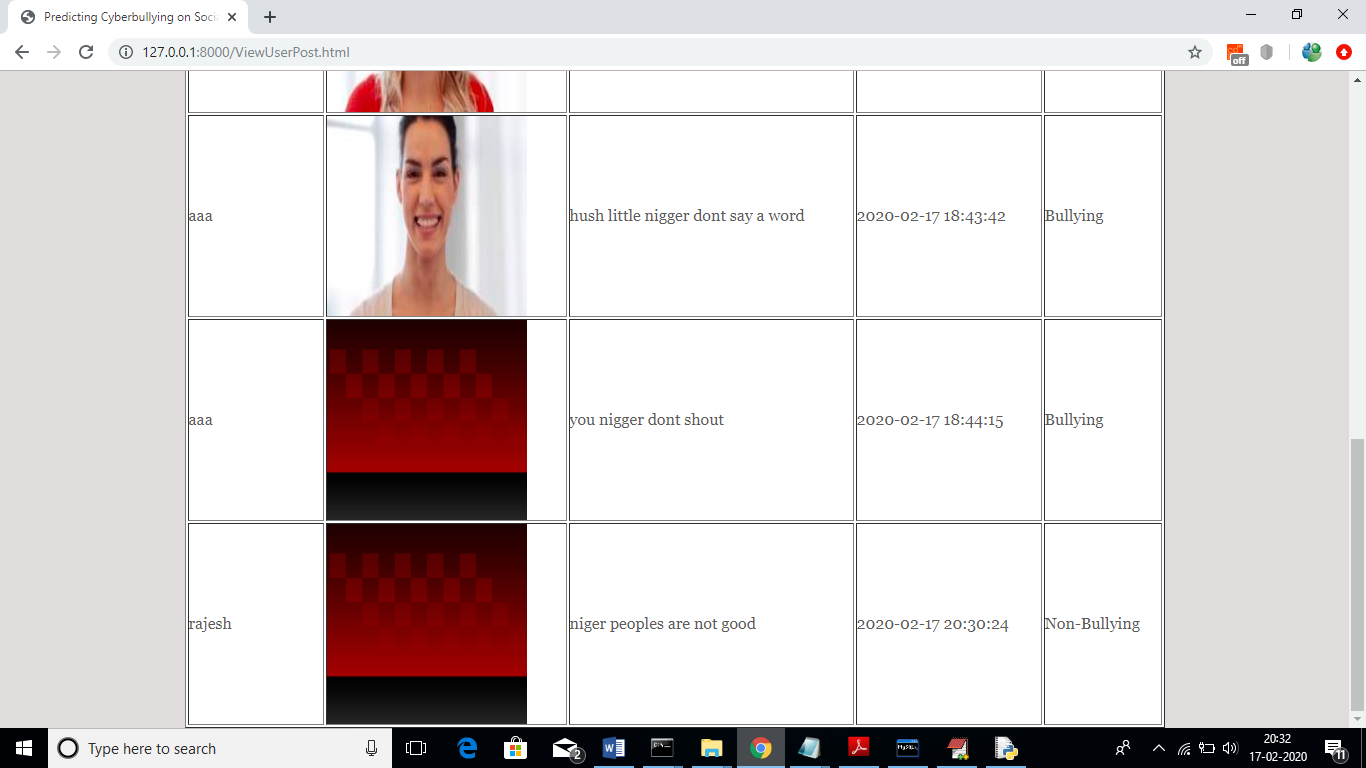


In above screen click on ‘Send Post’ link to get below screen



In above screen as post I added some messages and uploaded a photo also. After posting message will get below screen





In above screen we are seeing posts from all users and rajesh post predicted as ‘Non-Bullying’. Here based on words given in dataset will get prediction as bullying on non-bullying.

CONCLUSION

Social media and the internet have opened up new forms of both empowerment and oppression. Meaningful engagement has transformed into a detrimental avenue where individuals are often vulnerable targets to online ridiculing. A predictive model to detect this Cyberbullying in online content is imperative and this research proffered a prototype model for the same. The uniqueness of the proposed hybrid deep learning model, DLCNN is that it deals with different modalities of content,

namely, textual, and info-graphic (text with image). The results have been evaluated and compared with various baselines and it is observed the proposed model gives superlative performance accuracy. The limitations of the model arise from the characteristics of real-time social data which are inherently “high-dimensional”, imbalanced or skewed ‟heterogeneous”, and “cross-lingual”. The growing use of micro-text (wordplay, creative spellings, slangs) and emblematic markers (punctuations and emoticons) further increase the complexity of real-time Cyberbullying detection.

**REFERENCES**

1. Rosa, Hugo, et al. "Automatic cyberbullying detection: A systematic review." *Computers in*

*Human Behavior* 93 (2019): 333-345.

1. Cheng, Lu, et al. "Xbully: Cyberbullying detection within a multi-modal context." *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019.
2. Balakrishnan, Vimala, Shahzaib Khan, and Hamid R. Arabnia. "Improving cyberbullying detection using Twitter users‟ psychological features and machine learning." *Computers & Security* 90 (2020): 101710.
3. Cheng, Lu, et al. "Hierarchical attention networks for cyberbullying detection on the instagram social network." *Proceedings of the 2019 SIAM international conference on data mining*. Society for Industrial and Applied Mathematics, 2019.
4. Salawu, Semiu, Yulan He, and Joanna Lumsden. "Approaches to automated detection of cyberbullying: A survey." *IEEE Transactions on Affective Computing* 11.1 (2017): 3-24.
5. Balakrishnan, Vimala, et al. "Cyberbullying detection on twitter using Big Five and Dark Triad features." *Personality and individual differences* 141 (2019): 252-257.
6. Singh, Vivek K., Souvick Ghosh, and Christin Jose. "Toward multimodal cyberbullying detection." Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems. 2017.
7. Soni, Devin, and Vivek K. Singh. "See no evil, hear no evil: Audio-visual-textual cyberbullying

detection." *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018): 1-26.

1. Dadvar, Maral, and Kai Eckert. "Cyberbullying detection in social networks using deep learning based models; a reproducibility study." *arXiv preprint arXiv:1812.08046* (2018).
2. Gencoglu, Oguzhan. "Cyberbullying detection with fairness constraints." *IEEE Internet Computing* (2020).
3. Raisi, Elaheh, and Bert Huang. "Cyberbullying detection with weakly supervised machine learning." *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 2017.
4. Dani, Harsh, Jundong Li, and Huan Liu. "Sentiment informed cyberbullying detection in social media." *Joint European conference on machine learning and knowledge discovery in databases*. Springer, Cham, 2017.
5. Dani, Harsh, Jundong Li, and Huan Liu. "Sentiment informed cyberbullying detection in social media." *Joint European conference on machine learning and knowledge discovery in databases*. Springer, Cham, 2017.
6. Cheng, Lu, et al. "Session-based Cyberbullying Detection: Problems and Challenges." *IEEE Internet Computing* (2020).
7. Kumar, Akshi, and Nitin Sachdeva. "Cyberbullying detection on social multimedia using soft computing techniques: a meta-analysis." *Multimedia Tools and Applications* 78.17 (2019): 23973-24010.
8. Haidar, Batoul, Maroun Chamoun, and Ahmed Serhrouchni. "Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content." *2017 1st Cyber Security in Networking Conference (CSNet)*. IEEE, 2017.
9. Cheng, Lu, Ruocheng Guo, and Huan Liu. "Robust cyberbullying detection with causal interpretation." *Companion Proceedings of The 2019 World Wide Web Conference*. 2019.