A Project Report on

## Behavioral Analysis of Internet Messaging

**And Malicious activity**

**Detection.**

Submitted in partial fulfillment of the requirements

in

### Computer Engineering

by

### Ujjwal Jain (17102027)

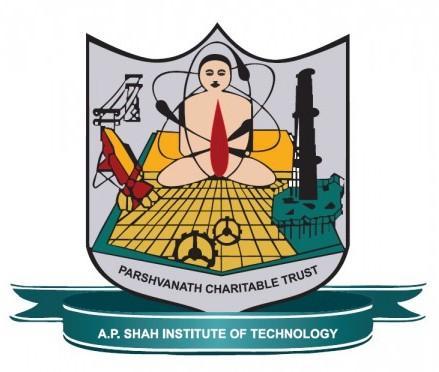
**Bhavik Jain (17102058)**

**Rishabh Mehta (17102039)**

**Shrinath Suryawanshi (17102012)**

Under the Guidance of

### Prof. Shafaque Syed



#### Department of Computer Engineering

#### A.P. Shah Institute of Technology G.B. Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

#### Academic Year 2020-2021

**Approval Sheet**

This Project Report entitled ***“Behavioral Analysis of Internet Messaging and Malicious activity Detection”*** Submitted by ***“Ujjwal Jain” (17102027), “Bhavik Jain” (17102058), “Rishabh Mehta” (17102039), “Shrinath Suryawanshi” (17102012)*** is approved for the partial fulfillment of the requirement in ***Computer Engineering Department*** from ***University of Mumbai***.

Prof. Shafaque Syed

Guide

Prof. S.H. Malave

Head, Computer Engineering Department

Place: A. P. Shah Institute of Technology, Thane

Date

### CERTIFICATE

This is to certify that the project entitled ***“Behavioral Analysis of Internet Messaging and Malicious activity Detection”*** submitted by ***“Ujjwal Jain” (17102027), “Bhavik Jain” (17102058), “Rishabh Mehta” (17102039), “Shrinath Suryawanshi” (17102012)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Science Department.***, to the University of Mumbai, is a bonafide work carried out during the academic year 2020-2021.

Prof. Shafaque Syed

Guide

Prof. S. H. Malave Dr. Uttam D. Kolekar Head, Computer Engineering Department Principal

External Examiner

Place: A. P. Shah Institute of Technology, Thane Date

### Declaration

We declare that this written submission represents our ideas in our own words and where others’ ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

———————————————

(Signature)

———————————————

(Ujjwal Jain, 17102027)

(Bhavik Jain, 17102058)

(Rishabh Mehta, 17102039)

(Shrinath Suryawanshi, 17102012)

Date

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | Project Conception and Initiation................................................................................ | | | 1 |
|  | **1.1** | Abstract........................................................................................................... | | 1 |
|  | **1.2** | Objectives........................................................................................................ | | 1 |
|  | **1.3** | Literature Review............................................................................................ | | 1 |
|  | **1.4** | Problem Defination…..................................................................................... | | 2 |
|  | **1.5** | Scope…........................................................................................................... | | 3 |
|  | **1.6** | Technology Stack…........................................................................................ | | 3 |
|  | **1.7** | Benefits for environment and society….......................................................... | | 4 |
| **2.** | Project Design.............................................................................................................. | | | 5 |
|  | **2.1** | Design (Flow of modules) ……………………………………….................. | | 5 |
|  | **2.2** | Class Diagram.…………………………........................................................ | | 6 |
|  | **2.3** | Modules…....................................................................................................... | | 6 |
| **3.** | Implementation………………………………………………………........................ | | | 10 |
|  | **3.1** | Proposed System……………………............................................................. | | 10 |
|  |  | 3.1.1 | Algorithms……………………….………………………………… | 10 |
|  |  | 3.1.2 | Pseudocode …................................................................................... | 14 |
|  |  | 3.1.3 | Platforms for execution……………………………………………. | 18 |
| **4.** | Results......................................................................................................................... | | | 19 |
| **5.** | Conclusion and Future Scope...................................................................................... | | | 21 |
| **6.** | References................................................................................................................... | | | 22 |
| **7.** | Bibliography................................................................................................................ | | | 23 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig 1.4** | Cyberbullying | 2 |
| **Fig 2.1** | Flowchart | 5 |
| **Fig 2.2** | Class Diagram | 6 |
| **Fig 2.4.1** | Dataset | 7 |
| **Fig 3.1.2.1** | Random Forest | 11 |
| **Fig 3.1.2.2** | Decision Tree | 12 |
| **Fig 3.1.2.3** | KNN | 13 |
| **Fig 3.1.2.1** | Preprocessing | 14 |
| **Fig 3.1.2.2** | Dataset Splitting | 14 |
| **Fig 3.1.2.3** | Random Forest | 15 |
| **Fig 3.1.2.4** | Decision Tree and KNN | 15 |
| **Fig 3.1.2.5** | Tf-Idf | 16 |
| **Fig 3.1.2.6** | Model Dumped in Pickle File | 16 |
| **Fig 3.1.2.7** | Flask display and import functions | 16 |
| **Fig 3.1.2.8** | Flask Predictions using ML | 17 |
| **Fig 3.1.2.9** | JavaScript code for client side | 18 |
| **Fig 4.1** | Classification Report | 19 |
| **Fig 4.2.1** | Confusion Matrix without normalization | 20 |
| **Fig 4.2.2** | Confusion Matrix with normalization | 20 |

**1. Project conception and initiation**

* 1. **Abstract**

Social media plays a very important role in everyone’s life. We can’t leave without it but it can be very dangerous for youth as they could become a victim of social trapping and psychological depressions. Internet messaging is having adverse effects on youth and the country. Misuse of “chatting” has become a common phenomenon and people are widely using it for some unusual activities. On social networking sites like Facebook, WhatsApp where teenagers communicate through explicit messages and trap the opposite gender which can be worst. In this paper, the methodology for dealing with such types of messages is discussed. Initially, text messages are processed through the classifier algorithm which would predict the behavior of the users, and the system will alarm if any malicious activity is detected.

* 1. **Objectives**
* To help the people who are unable to communicate with their closed ones about any threats or cyberbullying.
* To help the people from the threats or blackmailing
* To detect some unusual activities conducted via chats.
* To help the users from not getting into any unwanted activities which can mislead them.

**1.3 Literature review**

It considers one of the actual threats of the Internet of things related to the transmission of malicious messages between user. Methods of the transmitted data automatic verification and filtering by some set of features in Internet systems of things are analyze. It proposes an approach to the intelligent detection of unnatural conversation between user, which could minimize the likelihood of the implementation of threats.

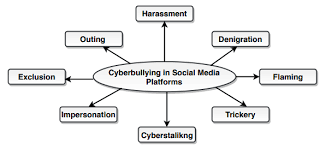
The major problem in Instant Messengers, much of sensitive and personal information, disclosed through socio-engineered text messages for which solution is proposed but, detection of words using chatting technique in Instant Messengers is not yet done which is the motivating factor to carry out the work. Online criminal's now-a-days adapted technique along with text message and wraps out personal information leads to threat and hindrance for privacy.

Psychological health disorders pose a growing threat to society. Disorders such as Depression, Post-Traumatic Stress Disorder (PTSD), and mild Traumatic Brain Injury (mTBI), are often under-diagnosed and under-treated. Crisis hotlines are often the last resort for people who, from the lack of proper treatment, are considering suicide or intend to harm themselves or others.

Harassment by cyberbullies is a significant phenomenon on the social media. Existing works for cyberbullying detection have at least one of the following three bottlenecks. First, they target only one particular social media platform (SMP). Second, they address just one topic of cyberbullying. Third, they rely on carefully handcrafted features of the data. We show that deep learning-based models can overcome all three bottlenecks. Knowledge learned by these models on one dataset can be transferred to other datasets. Our experiments provide several useful insights about cyberbullying detection. To the best of our knowledge, this is the first work that systematically analyzes cyberbullying detection on various topics across multiple SMPs using deep learning-based models and transfer learning.

**1.4 Problem Definition**

An outburst of Social networking pages, without proper monitoring, highly endangers personal security especially of teens and youth who are highly exposed to many inhumane activities. They get addicted to the internet Messaging over social networking sites presenting the serious threat of getting beguiled into extreme and even horrendous acts like Trapping, girl-Trafficking, motivation to become extremists, and other evils. Also, cyberbullying, blackmailing and threats given on social media can lead to anxiety, depression, and even suicide in some cases.



**Fig 1.4: Cyberbullying**

**1.5 Scope**

Cyberbullying is the use of electronic communication to bully a person by sending harmful messages using social media, instant messaging or through digital messages. It can lead to stress, anxiety, depression and even suicide in some cases. Same goes with the threats as people are ready to do anything that is inappropriate due to threats given to them. Also, sometimes blackmailing also leads to above problems and it is dangerous. It is very damaging to adolescents and teens. Also, once things are circulated on the Internet, they may never disappear, resurfacing at later times to renew the pain. So, to overcome these issues detecting the activity as malicious or not is very important which will help to stop cyberbullying, blackmailing and threats given on social media.

**1.6 Technology Stack**

**Languages:**

* Python: Python is high-level programming language and its use of indentation makes code easy for reading and its object-oriented approach helps to write clear and logical code.
* HTML/CSS: Html is the markup language use for Web pages and CSS is used for styling the html documents.
* JavaScript: JavaScript is scripting language mainly used for client-side web page. It helps in creating dynamically updated content, control media (images, audios, etc.).

**Techniques:**

* Random Forest Classifier: A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy.
* Pipeline: It is used to assemble several steps that can be cross-validated together while setting different parameters.
* TF-IDF: It is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.
* Flask: It is python web framework built with a small core easy to extend philosophy.It is based on Werkzeug WSGI toolkit and Jinja2 template engine.

**1.7 Benefits for environment and society**

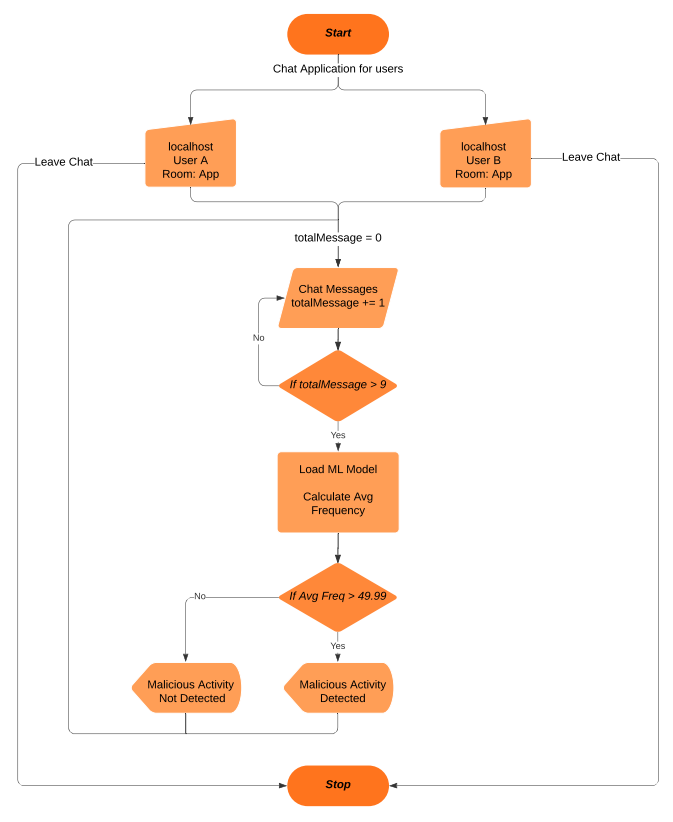
The idea of this project is concerned with the new generation as well as the present generation, this system will be very helpful for both of the generations. It will be mainly helpful for the women around us. We know and have knowledge about the cybercrimes and threats given on Social media and the victims are not able to express themselves due to many reasons. So, our system will send an alert notification to their responsible guardian on the behalf of the victim. So, they can get support from their guardians during tough times and we hope that our system will help in reducing Cybercrime rates.

**2.Project Design**

**2.1 Design (flow of modules)**

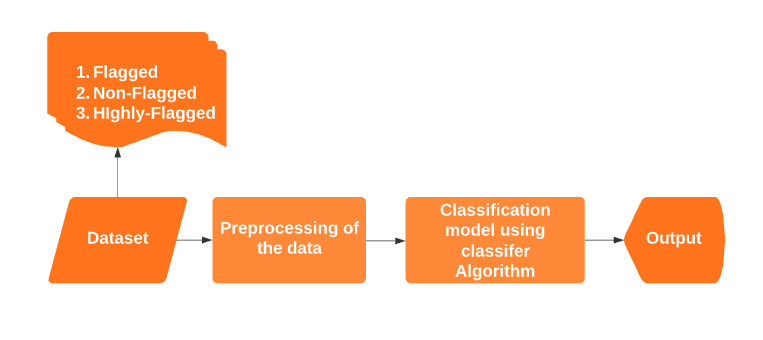
Initially User will get the chat application on their browser where they can chat with each other. They have to enter the username and room name in the input fields which will be used to validate and will provide the chatting interface to the user. Room name should be the same for the user to chat with each other else they will be able to chat but not with each other as they are logged in different rooms.

We have declared an empty list which will contain the message of the users. As soon as the length of the list is greater than 9 i.e. if the length is equal to 10 it will use the pickle file from which the ML Model will be loaded and it will predict the frequency of each sentences. Further it will calculate the average frequency which will be compared with the threshold frequency and if it is greater than threshold frequency so it will display malicious activity detected else it will display malicious activity not detected.



**Fig 2.1: Flowchart**

**2.2 Class diagram**



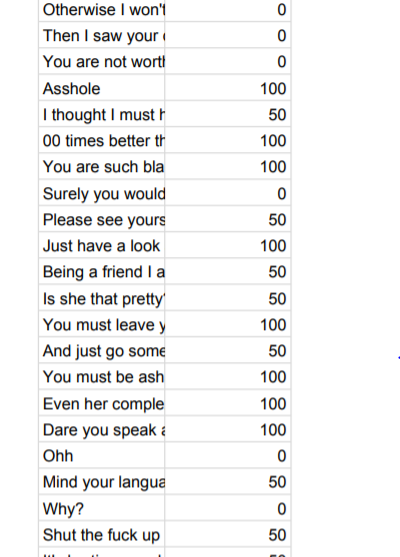
**Fig 2.2: Class diagram**

It shows that our dataset consists of three different labels as Non-Flagged with 0, flagged with 50 and Highly-Flagged with 100 as frequency. Further we’ll apply some preprocessing steps on the dataset and will remove unwanted character. Now, we’ll use classifier algorithm for training our dataset and based on that model will predict the output for the text messages passed to it.

**2.3 Module-1**

Data Gathering

* We had manually made the database by mutually chatting within ourselves.
* Also, we had taken the data from some online datasets available for free.
* Some curse/abusive words and their synonyms from the online tools.
* There are two columns in the dataset as Comments which are having the text messages and Frequency which is having the specific frequency provided to each sentence which belongs to flagged (50), Non-flagged (0) and Highly-flagged (100).
* Our Dataset contains total 35,523 sentences in which 10,205 sentences belong to Non-flagged, 14,417 sentences belong to Flagged and 10,901 sentences belong to Highly-flagged.



**Fig 2.4.1: Dataset**

**Module-2**

Preprocessing

* We had removed Stop words which are used for forming the sentences which are grammatically correct but it has no use in our ML classification model as it doesn’t add any meaning to the sentences.
* Also, we are keeping the data which has alphabetic characters and other extra characters, special characters as well as digits are removed from the dataset.
* All the data has been converted to lower case as mixed-case occurrences of the data can be sometimes not be able to effectively learn the data correctly and it provides different output for different words (e.g., ‘Canada’ vs. ‘canada’). This issue can be solved by lowercasing the whole dataset.

**Module-3**

Building ML Model Classifier

* For splitting the dataset into training and testing set we have used train\_test\_split function from sklearn model selection which will randomly split the dataset into 75:25 ratio i.e., 75% training and 25% testing dataset.
* Classifier: We have used Random forest classifier, decision tree classifier and K-nearest neighbors classifier for our model.
* A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy.
* Decision tree is supervised learning algorithm and it uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
* KNN algorithms use data and classify new data points based on similarity measures (e.g., distance function). Classification is done by a majority vote to its neighbors.
* Vectorizer: We have used TF-IDF which is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.
* We have defined a pipeline which consists of both classifier and vectorizer, which will be used to apply fit on training dataset. We have divided the features into 5 parts and we will call pipeline on these features. Now, we will calculate accuracy on different no. of features. We had used Random forest algorithm as it is providing the highest accuracy of 95.03% as compared to decision tree algorithm (93.47%) and K-nearest neighbors algorithm (44.95%).
* Classification Report on the actual and predicted labels gives the 95% of Precision, Recall as well as F1-Score. Precision shows the accuracy of positive prediction, Recall shows the fraction of positives that were correctly identified and F1-score is a weighted harmonic mean of precision and recall.
* Confusion Matrix shows that our model correctly predicted 2464 of 2531 Non-flagged sentences, 3448 of 3620 Flagged sentences and 2520 of 2730 Highly-flagged sentences.
* Finally, we have dumped our model to a pickle file which is used to serialize a python object into a binary format which will be used in our flask app for prediction of the frequencies which can provide us the specific output.

**Module-4**

Predicting the Output using Flask

* Flask-SocketIO: It gives flask applications access to bi-directional communications between the clients and the server. Client-side application has been developed using JavaScript.
* Flask-Session: It is used for storing the data on the server. The session is the interval at which the client logs on to the server and logs out the server. The data that is required to be saved in the session is stored in a temporary directory on the server.
* Initially we have loaded the ML Model using the pickle load function and we are creating the flask instance object. We have used session type as filesystem for storing our text message on folders and not on the server. We have created 5 functions for our flask application which are as follows:
* Index (): Used for rendering the index page of our application.
* Chat (): It will take the username and room from the form and will store that in the session for which user has logged in and it will render the html page for chatting.
* Join (): It will get the room name entered by the user and it will display the message as “user has entered the room”.
* Text (): It will display the chat message of the user in the text field along with the username of the user who is sending the message. Further it will check the length of the list and if it is greater than 9 i.e., if it I equal to 10 it will use the ML model for predicting the frequencies of each message and it will calculate the average of it. If the average is greater than 49.99, it will display Malicious activity detected else it will display malicious activity not detected. Further will delete the message from the list and it will start storing the new messages and so on. This process will continue till the user leaves chat.
* Left (): As the user leaves the chat it will clear the session in which username and room are stored and will display the message as “user has left the room”.

**3. Implementation**

**3.1 Proposed system**

* As we are seeing on social media and news that many young adults are becoming the victim of blackmailing, cyber bullying and threats given on social media platforms.
* So, this problem encourages us to make this system as our system will compare the message retrieved from the online chatting with the already created database.
* The main advantage of our system is that our system will detect the meaning of chats using the frequency of curse/abusive words used in that chat and it will alert the respected guardian that your child is in some trouble.
* It will help in reducing suicide rates as victim is not able to tell their guardians about their problems in personal life.
* Also, victims are threatened for not involving their loved ones, so this system will help in that case.
* This cyber-trapping which is very difficult to catch can be easily handled using our system.

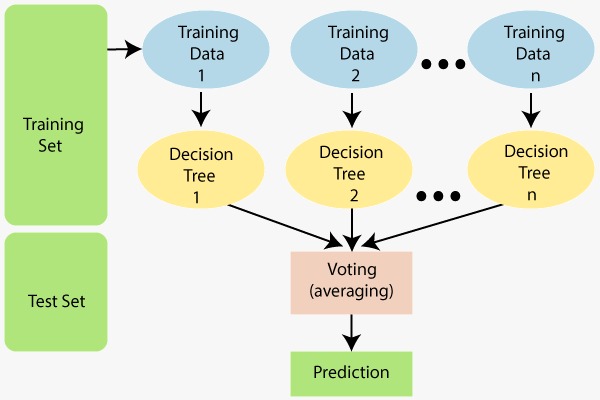
**3.1.2 Algorithms**

Random Forest Algorithm

Random forest is a [supervised learning algorithm](https://builtin.com/data-science/supervised-learning-python). The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. **It builds multiple decision trees and merges them together to get a more accurate and stable prediction.** One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).



**Fig 3.1.2.1: Random Forest**

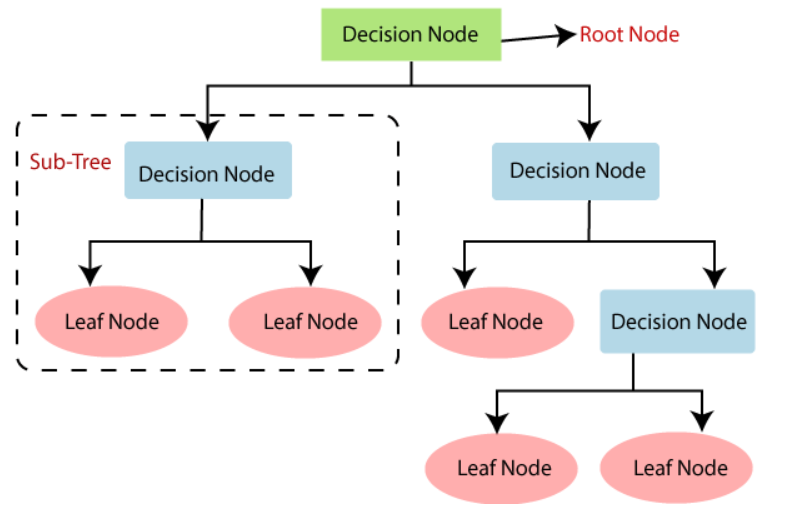
Decision Tree Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



**Fig 3.1.2.2: Decision Tree**

K-Nearest Neighbours (KNN) Algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm.

It can be used for Regression as well as for Classification but mostly it is used for the Classification problems. It is a non-parametric algorithm, which means it does not make any assumption on underlying data.

At the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



**Fig 3.1.2.3: KNN**

TF-IDF Algorithm

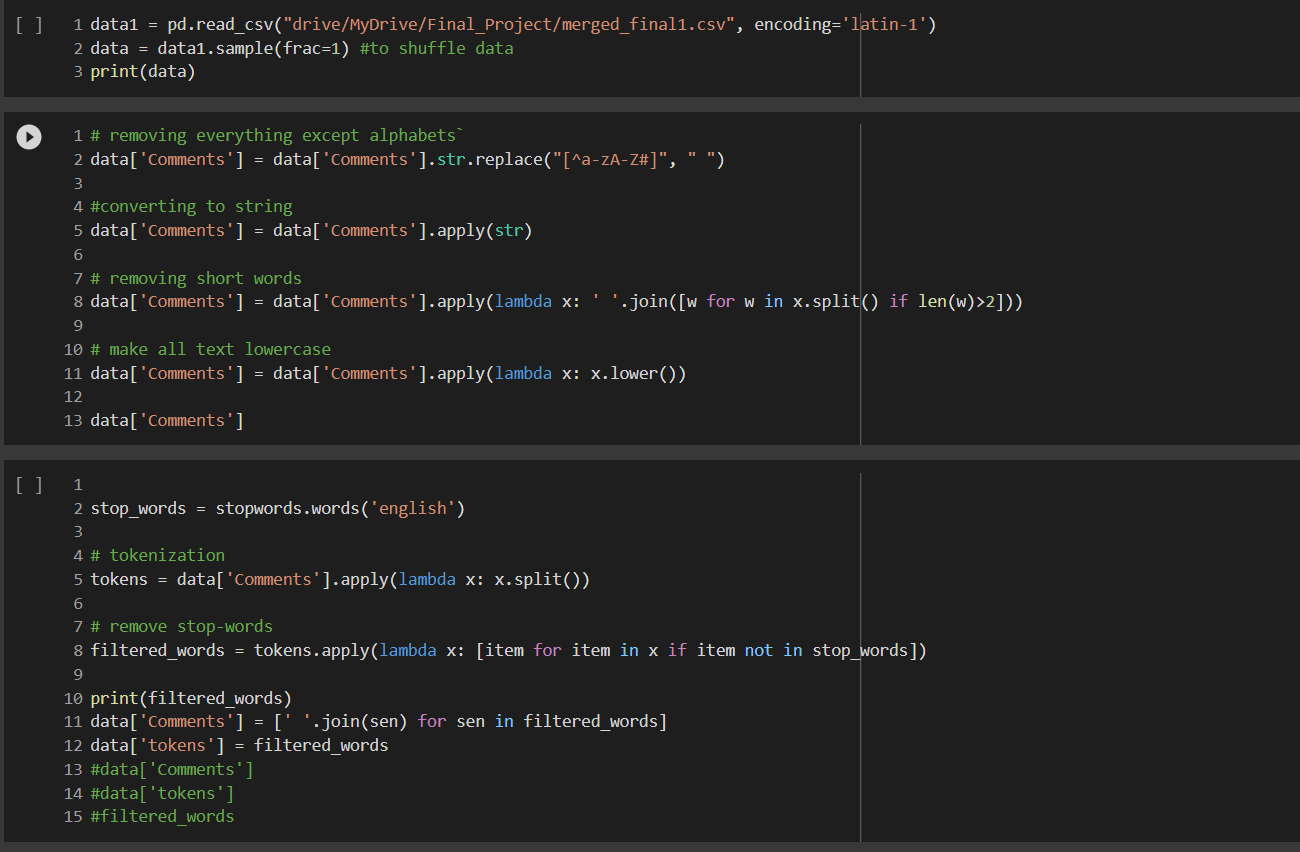
TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP).

It was invented for document search and information retrieval. It works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don’t mean much to that document in particular.

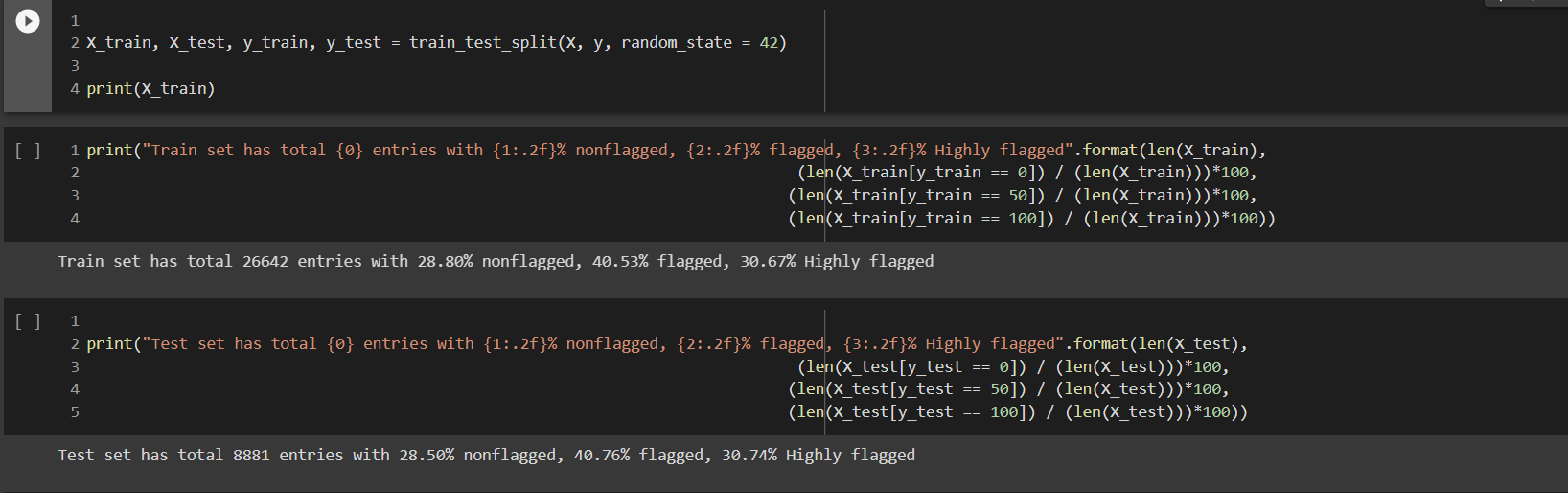
Term Frequency: It is the measurement of how frequently a term occurs within a document. The easiest calculation is simply counting the number of times a word appears.

Inverse Document Frequency: Inverse Document Frequency (IDF) is a calculation often used in conjunction with Term Frequency. The problem with term frequency is that frequent terms aren’t always the most important. IDF is calculated by dividing the total number of documents by the number of documents in the collection containing the term.

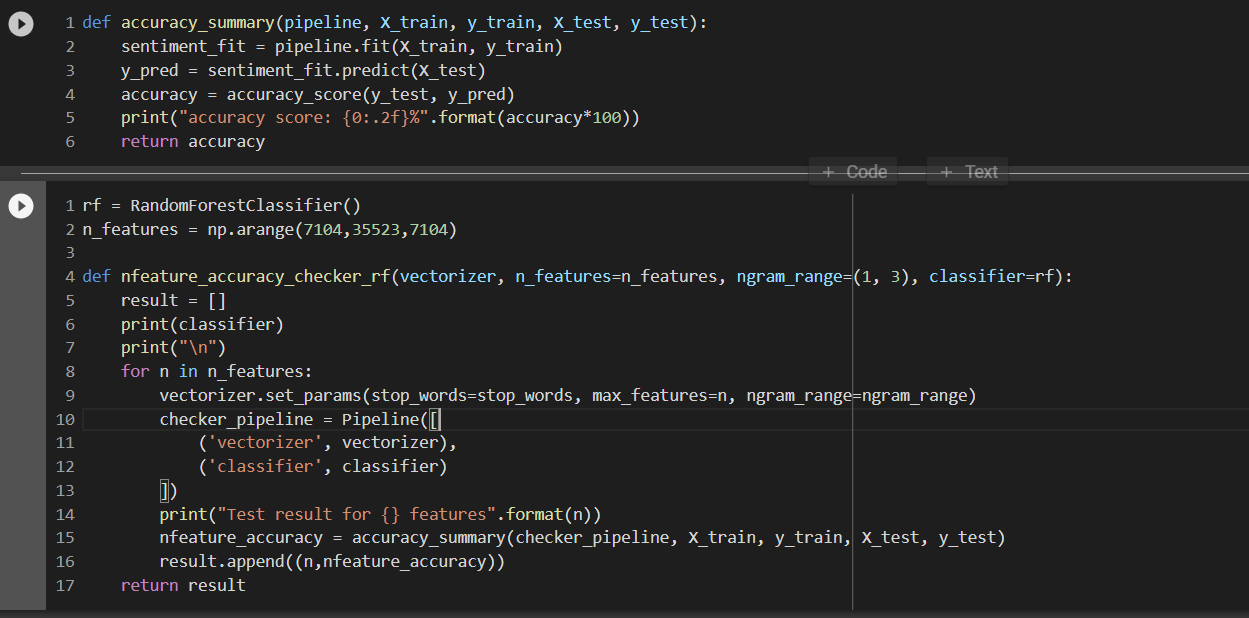
**3.1.3 Pseudo code**



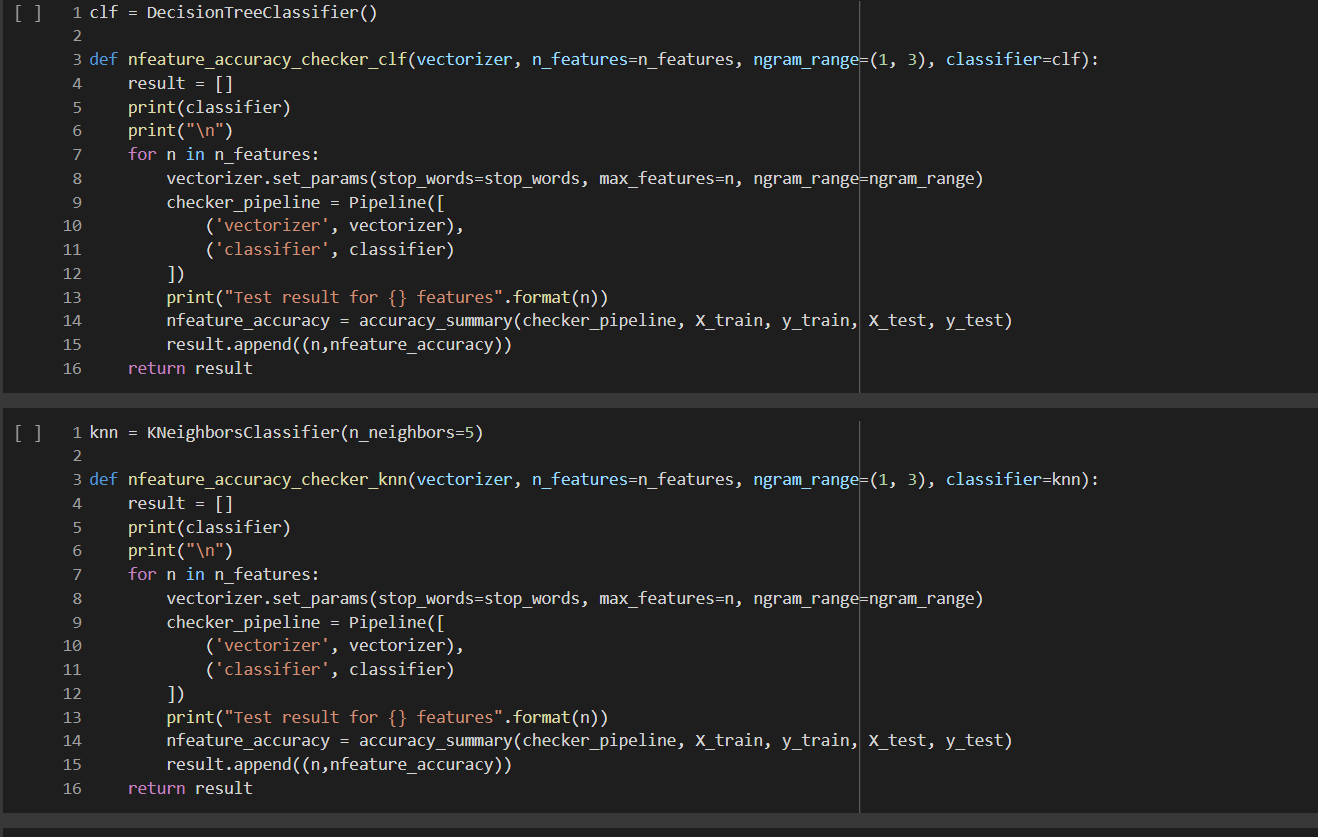
**Fig 3.1.3.1: Preprocessing**



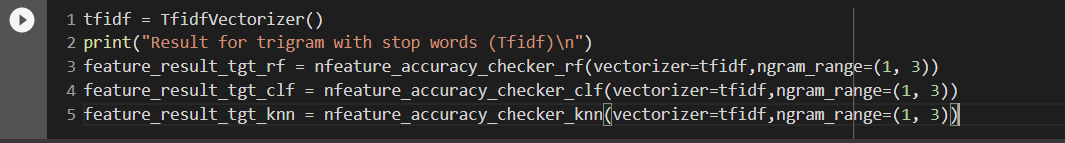
**Fig 3.1.3.2: Dataset Splitting**



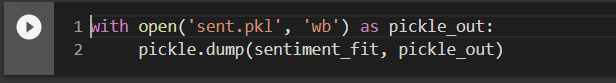
**Fig 3.1.3.3: Random Forest**



**Fig 3.1.3.4: Decision Tree and KNN**



**Fig 3.1.3.5: TF-IDF**



**Fig 3.1.3.6: Model dumped in Pickle file**



**Fig 3.1.3.7: Flask imports and display functions**



**Fig 3.1.3.8: Flask Prediction using ML**



**Fig 3.1.3.9: JavaScript Code for Client-Side**

**3.1.4 Platforms for execution**

Google Colaboratory

It is a product from Google Research. Colab is basically a free Jupyter notebook environment running wholly in the cloud. Most importantly, Colab does not require a setup, plus the notebooks that you will create can be simultaneously edited by your team members – in a similar manner you edit documents in Google Docs. The greatest advantage is that Colab supports most popular machine learning libraries which can be easily loaded in your notebook.

Localhost for Browser

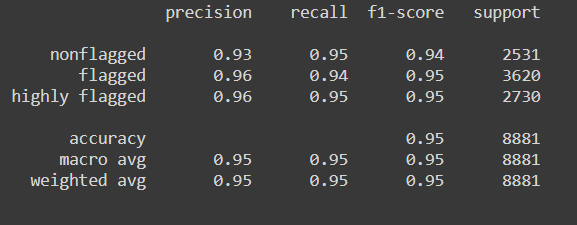
It is the default name used to establish a connection with your computer using the loopback address network.

The loopback address has a default IP (127.0.0.1) useful to test programs on your computer, without sending information over the internet. This helps when you are testing applications that aren’t ready for the world to see. When you call an IP address from your computer, you usually try to contact a different computer over the internet. However, with the loopback address, you are calling the localhost, aka your computer.

**4. Results**

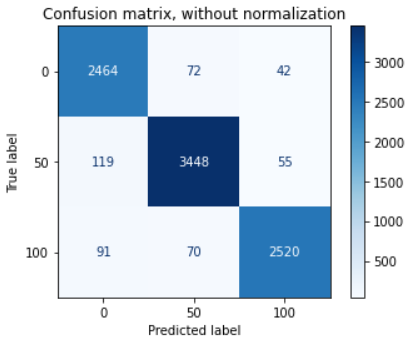
|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Algorithm** | **Accuracy** |
| 1. | Random Forest | 94.82% |
| 2. | Decision Tree | 93.47% |
| 3. | KNN | 43.32% |

RandomForest achieves highest accuracy among all the classifier algorithm. It gives accuracy of 94.82% as compared to 93.47% by decision tree and 43.32% by KNN. So, we have used random forest for generating our classification report and also for predicting the output.



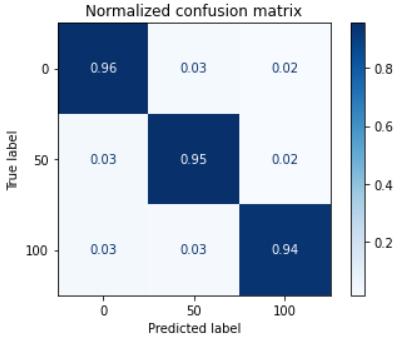
**Fig 4.1: Classification report**

Thisis the Classification report of our model which shows average precision, average recall and average F1-score as 95%.



**Fig 4.2.1: Confusion Matrix without normalization**

Confusion Matrix shows that our model has correctly predicted 2464 of 2531 Non-flagged sentences, 3448 of 3620 Flagged sentences and 2520 of 2730 Highly-flagged sentences.



**Fig 4.2.2: Confusion Matrix with normalization**

It is normalized confusion matrix which the accuracy of our model i.e., total percentage of sentences correctly predicted.

**5. Conclusion & Future Scope**

**Conclusion**

Our system is used for the betterment of the People and it is widely used for young age who are more prone to involve in these activities. It will give a chatting interface for the users where several users can chat with each other having a common room. From that text field, we will extract the message and if the count of the message is 10 so it will apply the ML model and predict the frequency for each sentence. Finally, it will calculate the average of all the frequencies and will be compared with the threshold frequency i.e., we have set the threshold frequency as 49.99. If the average frequency is greater than 49.99 it will display malicious activity detected else it will display malicious activity not detected. For the classification model, we have used a random forest classifier which creates no. of decision trees using random samples and based on voting it gives output.

**Future Scope**

We’ll create the system where the user needs to log in using his contact number and also need to enter the guardian contact number where we can send the notification as some malicious activity is there. Also, the GUI of the chatting system will be made attractive as well as responsive for all the window sizes and will add more functionality where users can share images and other media like other chatting systems.

As of now, we are running our chatting system on localhost, but in the future, we’ll try to convert our application to a website as well as a mobile app that can be directly downloaded from the play store and can be used for chatting.

**6. References**

[1] Behavioral analysis of internet messaging and malicious activity detection, K. Srividya, A. Mary Sowjanya, 2016 International Conference on Advances in Human Machine Interaction (HMI), Doddaballapur, India

[2] Deceptive phishing detection system: From audio and text messages in Instant Messengers using Data Mining approach, Mohammed Mahmood Ali and Lakshmi Rajamani, International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012) , Tamilnadu, India

[3] Automatic search of the malicious messages in the internet of things systems on the example of an intelligent detection of the unnatural agents requests, Anastasia Iskhakova, Roman Meshcheryakov, 2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC), Vladivostok, Russia

[4] Automatic detection of psychological distress indicators and severity assessment in crisis hotline conversations, Maciej Pacula, Talya Meltzer, Michael Crystal, Amit Srivastava, Brian Marx, 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy

[5] Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms, Sweta Agrawal, Amit Awekar, Accepted for ECIR 2018

[6] Cyberbullying Detection: A survey on Multilingual Techniques, Bataul Haidar, Maroun Chamoun, Fadi Yamout, 2016 European Modelling Symposium (EMS), Pisa, Italy.

[7] Cyberbullying classification using text mining, Noviantho, Sani Muhamad Isa, Livia Ashianti, 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia.

[8] Detection of Cybrbullying using Deep Neutral Network, Vijay Banerjee, Jui Telavane, Pooja Gaikwad, Pallavi Vartak, 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India.

**7.Bibliography**

[1] Sentiment Classification with Python, Susan Li, Sep 8, 2018, Toronto, Canada.

[2] Random Forest Algorithm-An overview, Great Learning Team, Feb19, 2020

[3] Getting into Random Forest Algorithms, Nibin, April 30, 2021.

[4] Split your dataset with scikit-learn, Mirko Stojiljkovic, Nov 23, 2020.

[5] Decision Tree Algorithm for Classification. Akshay Sharma, Feb 25, 2021.

[6] Introduction to K-Nearest Neighbours, Tavish Srivastava, March 25, 2018.

[7] Tf-idf Vectorizer scikit-learn, Mukesh Chaudhary, April 20, 2020.

[8] Natural Language processing feature engineering using tfidf, Cory Maklin, May 5, 2019.

[9] Understanding the Classification report through sklearn, Muthu, July 7, 2018.

[10] Understanding Confusion Matrix, Sarang Narkhede, May 9, 2018.

[11] Chi-square test for feature selection in machine learning, Sampath Kumar Gajawada, Oct 4, 2019.

[12] Understanding Python Pickling with example, Omkar Pathak, Nov 13, 2018.

[13] An introduction to the flask python web app framework, Nicholas Hunt-Walker, April 2, 2018.

[14] Building your first chat application using flask, Samhita Alla, May 26, 2018.

[15] Building a multi-room web chat application with flask, Miguel Grinberg, Nov 17, 2020.