

ABUSIVE COMMENT DETECTION AND PREVENTION

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

Seeni Harish (Reg. No. 37110693)

Savaram Kishore (Reg. No.37110692)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC JEPPIAAR NAGAR, RAJIV
GANDHI SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **SEENI HARISH (Reg. No. 37110693)** and **SAVARAM KISHORE (Reg. No. 37110692)** who carried out the project entitled "**ABUSIVE COMMENT DETECTION AND PREVENTION**" under my supervision from **August 2020 to March 2021**.

Internal Guide

Mrs. K.Indra.M.Tech.,

Head of the Department

Dr. S. Vigneswari,M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Seeni Harish** and **Savaram Kishore** here by declare that the Project Report entitled “**ABUSIVE COMMENT DETECTION AND PREVENTION**” is done by me under the guidance of **Mrs.K.Indra.M.Tech.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. K.Indra.M.Tech.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Antisocial and abusive conduct has become more common because of the emergence of social media. Reputation of perspective-rich online tools such as feedback forums and micro blogging platforms has enabled users to identify and express their opinions in real time all around the globe. It also ends up in people who use hateful language to post ridiculous and violent material online. Those are often aimed to show their opposition to a personal or culture. So, it is critical for legislators and social networking sites to identify hate speech and prevent the proliferation of any unlawful conduct. The character of structures of the tongue makes this task even more tedious. Being a challenging task, we are fascinated to propose a machine learning system supported Decision tree to classify Hate Speech, Offensive speech, Profanity or not. we've got done experiments with English dataset. Dataset is employed to coach the model for performing various actions. it's a group. Classification of a given input into HOF or NOT is completed on the idea of count of numbers of bad words which is computed by comparing each token with the respective word present within the predefined set of abusive words using decision tree.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE No
1	ABSTRACT	V
	INTRODUCTON	1
	1.1. INTRODUCTION	
	1.2 PROJECT CATEGORY	3
	1.3 OBJECTIVES	3
2	LITERATURE SURVEY	4
	Literature survey	4
	2.1SOFTWARE REQUIREMENT	
	Technical Feasibility	7
	Economic Feasibility	8
	Operational Feasibility	8
3	Aim and Scope	
	Aim	4
	Scope	4
	Design approach and Architecture	10
4	MODULE DESCRIPTION	11
	4.1 Reading The Data	11
	4.2 Pre-Processing	12
	4.3 Special Characters Removal	13
	4.4 Classification	14
	4.1.1 IMPLEMENTATION,TESTING,MAINTENANCE	
	Introduction to languages	23
	Introduction to ides	24
	System implementation	24
	Programming language	25
	Introduction to technology	25
	Introduction to tools used	25
5	RESULTS AND DISCUSSION	
	5.1 Snapshots of the project	27
	5.2 Brief Description of various modules of system	

6

Text collection	29
Text-processing	29
Text Transformation	29
Feature Selection	29
Data mining	29
Interpretation and Evaluation	29

5.3 Snapshots of folder Structure	35
-----------------------------------	----

CONCLUSION AND FUTURSCOPE

6.1 CONCLUSION	36
6.2 FUTURE SCOPE	36
6.3 REFERENCE	37

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE
NO		
1	A comment section in Instagram	2
2	Determination of polarity scores	13
3	A sample cat dataset	15
4	A decision tree of the above cat dataset	16
5	A modal of confusion matrix	17
6	A graph illustrating precision and accuracy	19
7	Formulas for calculating precision, recall, accuracy	
8	Formula for calculating the precion	21
9	Formula for calculating the recall	21
10	Formula for calculating F-score	22
11	A raw dataset of text from twitter	30
12	Data preprocessing	31
13	Snapshot 1 of model anaconda jupyter notebook	
14	Data Transformation	31
15	Feature section phase –1	32
16	Feature section phase -2	33
17	Data mining	34
18	Interpretation	

1.1 INTRODUCTION TO THE PROJECT

In social media, anyone is free to post their ideas and views while not declaring his/her identity. Detection of Hate Speech, Offensive language and utterance employed by social media users became one in every of the foremost challenges of this case. Social media users were being targeted by Hate Speech and Offensive language like abusive, hurtful, disparaging or unlawful user-generated content by some mischievous users. These on-line platforms offer academic degree open place to debate and treat totally different matters however abusive comments and on-line violence on people have turned this into a awfully necessary social issue. As a result of the misuse of on-line interactions, an outsized range of people have fallen into depression, anxiety, and alternative psychological state issues.

The worldwide use of social media is ever growing. The 2019 social network figures indicate that there are 3.5 billion consumers of social media globally, and that number is scarcely increasing. That's equal to 40 quarter of this country. With the growth of social media, users particularly teens are increasingly disbursing significant sums of it on various social networking platforms to communicate with others, exchange data, and follow mutual interests. In 2011, seventeenth of adolescents using commonplace social networking sites and almost one in four teenagers reach their top social network sites ten as more times regularly. In fact, cyber-bullying happens through the social media indicate derogatory comments. This has been noticed that nineteen teenagers mention someone on social networking platforms having published or denoted negative or humiliating stuff about them. When teens become loads of relatively derogatory filled with skewed and dangerous material relative to adults, web monitoring of offensive content to secure teen health web is an essential activity for academic degree. Social network managers usually conduct a manual web analysis on-line contents to seek out and delete offensive material.

The internet is a hurling spot too, full of interesting data and stuff to learn and speak. But for some, it would be a dark and frightening place to rule over their violence of influential and unknown trolls victimizing the numerous social networking sites. Last year, a study commissioned by the Bench Research Centre showed that 73% of adolescents witnessed someone abused in how on-line, with 40% witnessing harassment on the net directly. Celebrities and celebrities have removed their social media pages in recent years as they get criticism on their statements about violence. The regulation of social media has been headlining news this year. And one in every of the continual themes has been the potential for AI to be used as a filter or safety mechanism to help the varsity giants' police the content revealed on their platforms. this may be a step within the correct direction. and since the key platforms, Instagram enclosed, come underneath increasing pressure to manage you'll expect to check lots of constant. When you're facing abuse on-line, it'll feel overwhelming and you may not confirm how to report an issue like this. Here's a handy orient how to change abuse on the foremost social media platforms and therefore the thanks to report incidents and therefore the thanks to fully block such comments.



Figure 1: A Comment section in Instagram

The main aim of this project is to prevent one of the ways of abusing people i.e., abusing people with comments on their individual posts that they share in some social media platform. This will help some of the people who are abused through comments to their posts or pictures in social media by completely blocking the abusive comments and it doesn't block the users at all. So, this might be useful in a respective manner.

1.2 PROJECT CATEGORY

This project is categorized under Research based projects. The research-based project is one during which the first goal is to accumulate knowledge of some kind or to resolve some quite uncertainty instead of producing a tangible product as a deliverable.

A characteristic feature of the research is obvious definition of the ultimate outcome of the project. The results of the research can cause useful results which can indicate that individual management plans are unreasonable or impracticable. Research projects are undertaken so as to accumulate new knowledge, while projects within the classical sense, are geared toward concrete results which will be products or services which may be sold on the market.

Already lot of research has been done on detection of Hate, Offensive and Profinite in Indo- European languages. many of us has suggested various methods to unravel the matter of detecting abusive comments. In most of the cases they're using character degradation and stop words method to unravel this problem but here we've used text based talking language so as to unravel this problem and here we've used decision tree classifier so as to conclude desired results.

1.3 OBJECTIVES

Detecting hate speech for lawmakers and social media platforms to discourage occurrence of any wrongful activities. In order to stop an individual or community to show their dissent.

To cease the misuse of online interactions and saving people falling into depression, anxiety, and other mental health problems. To discourage antinationalism as the part of eliminating disputes on the basis of caste, religion, races, nation etc.

To stop violence and letting people to practice stricter conventions of using social media as a platform of delivering their use publicly by preserving self-respect of others.

CHAPTER – 2, LITERATURE SURVEY

Kirti Kumari and Jyothi Prakash in HASOC 2019 detected hate speech and abusive content using Deep Learning approach. The Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) organizer team defines Hate Speech as describing negative attributes or deficiencies toward groups because of race, political opinion, sexual orientation, gender, social status, health condition or similar [10]. A large number of works [1,5,6,15] are reported by the researchers on Hate Speech detection for English language only and very few works [2,8] have been reported for mixed languages such as English-Hindi, English-Bengali, and other languages.

In this paper, they have used the multi-lingual HASOC Corpus [10] and proposed a deep learning model based on Convolutional Neural Networks (CNN) to identify Hate Speech and Offensive content on multi-domain social media platforms collected from Facebook and Twitter. They have used three types of embedding of text and transliteration tools to normalize Devanagari to Roman script for code-mixed Hindi corpus.

The rest of the paper is structured as follows. Section 2 presents associated works for the detection of Hate Speech and Offensive Language while Section 3 presents our suggested framework for identification of Hate Speech, Offensive language, and Profanity. Section 4 presents the finding of the suggested scheme. Finally, in Section 5, they have concluded the paper and have discussed the future directions for these tasks.

She described as social media and online platforms have grown in terms of impact and acceptance of users, various problems such as Hate Speech, Profanity and Offensive language on these platforms have increased drastically. Several systems have been proposed by researchers for automatic detection and classification of these problems. A lot of research works have been done for the English language but very few works have been done for the other languages and code-mixed languages. In this paper, we have focused on multi-lingual text,

especially for Hindi and English languages and used a deep neural network model to detect Hate Speech, Offensive language, and Profanity.

The Hindi-English code-mixed corpus was created by Aditya Bohra, Deepanshu Vijay and Vinay Singh using the tweets written online in the last five years. Tweets have been discarded from Twitter then using Twitter Python API which uses twitter's sophisticated search option. They've extracted the tweets by picking from politics other hashtags and keywords. Community marches, demonstrations, and so on, who have a strong reputation for hate speech. They accessed 1,12,718 tweets in json format from Twitter, which contains details such as timestamp, URL, email, person, re-tweets, comments, full name, I d and likes. A thorough review to delete all the disruptive messages was carried out.

A noted corpus of Hindi-English code-mixed content, consisting of tweet ids and the accompanying annotations, is provided in this article. They also introduce the controlled method used in code-mixed text to identify Hate Speech. The corpus is made up of 4575 code-mixed messages with annotated hate speech and regular expression. The vocabulary in the tweets is often annotated with the term root words. Word n-grams, phrase n-grams, punctuations, terms of negation and lexicon of hatred are the characteristics included in our classification scheme. Excellent 71.7 percent precision is reached when all characteristics are integrated into the task vector using SVM as the classification scheme. The corpus may be annotated as part of potential research with part-of - speech tags at word level and can produce improved outcomes. In addition, in particular, the annotations and studies outlined in this paper can also be performed for code-mixed texts comprising more than two languages from multilingual societies.

In the case of Support Vector Machine and Random Forest Classifier respectively, Support Vector Machine does stronger than Random Forest Classifier and achieves 71.7 percent maximum accuracy when all attributes have been used. Character N-Grams was found to be the most effective in SVM while Word N-Grams was the most reliable in Random Forest Classifier.

Warner and Hirschberg (2012) identified hate speech designed for different factors like ethnicity. Throughout their research, they described hate speech and then collected data from Yahoo and the American Jews Congress (AJC), that Yahoo received its newsgroup data, and AJC gave URL identified as offensive websites. Throughout their first attempt, they categorized data at paragraph

stage, but then used this data set for annotation by requiring annotators to manually transcribe the data collection. We concentrated on stereotyping and therefore agreed to allow stereotyping language model to label hate speech. They first created a classifier of an anti-Semitic expression. They found 9000 paragraphs which suited their normal speech and eliminated those subsections which were not provocative. Then seven more groups were chosen to annotate the results. They utilized two-fold cross validation classifier after that annotation for their gold corpus to find a polished collection of results.

Motivated by the research performed in Kwok and Wang, (2013) suggested a way of identifying black-hatred speech via Facebook. Thousands of tweets were organized to examine keywords or thoughts suggesting speeches of hatred. A test has been circulated to students of various races to determine the seriousness of the claims. A testing data collection of 24582 tweets was pre-processed to fix variance in pronunciation, delete terms from stoppage and eliminate URL etc. To identify tweets, NB classifier illustrated racial and non-racist messages, and they described influential characteristics from such tweets. The classifier reported 86 per cent precision.

To order to identify antagonistic posts on Twitter, Burnap et al. (2013) established a rule-based methodology and used relation terms as attributes. It often incorporated charges and forms of reference directed at an individual or individuals after a socially destructive incident as attributes, in an attempt to convey the meaning and usage of the word. Their findings indicated that traditional learning methods had changed.

With the support of social network and text mining research, Ting et al. (2013) suggested framework for detecting hate groups over Twitter. We also extracted functionality like keywords which are sometimes found in classes. Sureka et al. (2012) suggested a method to uncover hate promotion posts, consumers and their secret groups on YouTube focused on the data analysis and social network analytics. Chen et al. (2013) developed a system for the detection of YouTube terrorist images. Author derived user-generated data from lexical, syntactic, and content-specific elements, and used different feature-based classification techniques to identify images. Agarwal and Sureka also suggested a based crawler (best first scan and shark hunt) method for extracting accounts of YouTube users who spread hatred and extremism.

Different solutions to machine learning were made to tackle the toxic language issue. The bulk of methods deal with extraction of features from the document. Any research used lexical tools such as dictionaries, and bag-of-words. Such features were found to struggle to grasp the meaning of the sentences. Approaches based on N-gram have also been used which display comparatively better performance.

While lexical features work well in identifying offensive items, they struggle to discern offensive sentences that include the same terms but in specific orders without considering the syntactic structure of the entire sentence. The natural language process parser, introduced by Stanford Natural Language Processing Group, has been used in the same analysis to catch the grammatical dependencies inside a phrase.

Linguistic tools such as parts-of - speech were often used in hate speech identification issues, these methods consist of identifying the phrase type, for example, personal pronoun (PRP), Verb non-3rd person singular present form (VBP), Adjectives (JJ), Determiners (DT), Verb base types (VB).

Many experiments have been carried out on sentiment-based approaches to identify the offensive language that has been released in recent years. One illustration is the work utilizing emotion analysis to spot abuse in messages, and using subject models from the Latent Dirichlet Allocation (LDA) to classify important topics in such texts.

Research for the Identification of Violence were also performed on Cloud 2.0. Recent times, dispersed representations of terms, often called word embedding, have been suggested for a specific reason. Current usage of deep learning strategies is made using paragraph2vec methodology in text recognition and sentiment analysis. Convolutionary Neural Network (CNN) enabled classification, which applies to the development of a text classification CNN, where they worked with a Twitter hate-speech content classification method built on a deep-learning, CNN model. Our methodology is different from the aforementioned analysis as much of the research conducted on hate speech is on a particular subject or area based on the phenomenon that is emerging / developing in a defined time. At the time they attempted to concentrate various topics on specific research such as culture, sex, ethnicity etc. We have provided an approach in this research to

concentrate on basic queries. Secondly, our methodology varies from above as we use real-time stream tweets as well as details regarding user profiles.

CHAPTER – 3, Aim and Scope of the present investigation

Aim:

The main aim of this project is to prevent one of the ways of abusing people i.e., abusing people with comments on their individual posts that they share in some social media platform. This will help some of the people who are abused through comments to their posts or pictures in social media by completely blocking the abusive comments and it doesn't block the users at all. So, this might be useful in a respective manner.

Scope:

Text messaging is already a significant means of personal and corporate collaboration through the Online or cell phones. At just about the same time, flames (such as rants, taunts, and squalid phrases) are provocative / insensitive terms even for a wide range of reasons may target or offend the users. A common technique will be an automated racist or discriminatory program with a resistance factor for the identification of flame or offensive words. While a person may identify certain kinds of pointless frustrating materials among the interesting ones, computer software doesn't have an easy job.

Pattern classification and machine learning technologies have recently been around a multitude of Natural Language Data processing. It's like we have faced interact through text messages where there are a series of threats or offensive remarks (e-mails or various forms of texts). This would be good to provide an automated intellectual program to identify flames or any other offensive words, that could save time and attention for its subscribers. Inflammatory words may ridicule or mock someone or a class of people (assaults in a tirade such as violence towards certain class, society particular group, sex or philosophy).

Chapter – 4, Experimental or materials and methods; algorithms used
Project is the opening for every engineering product or device within the production process. It will be defined as' the process of applying various

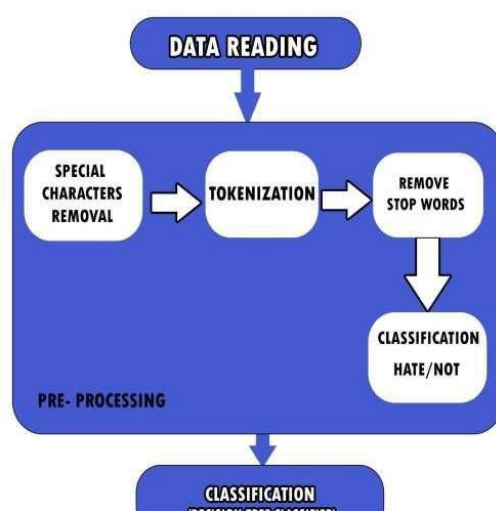
techniques and principles for the purpose of defining a device, or a method with insufficient detail to permit its physical realization.

Structure of the software design system and operational specifics are checked and recorded. System design may either be presented from a technical point of view or is an iterative process by which specifications are converted into a "blue print" for the software construction. Planning is defined at a high level of abstraction, a level that will be translated directly into the selected data, functional and behavioural requirements. Preliminary design is worried the transformation of requirements into a knowledge and software architecture.

Software design sits at the software development process technological kernel and is implemented irrespective of event paradigm and application area. Application is that the opening for any engineered device or system within the production process. The aim of the designer is to provide a model or representation of an object that may be built later. Starting with the system requirement being defined and evaluated, system design is the first of the three technological activities — design, code, and test the software needs to be developed and tested.

One word "Efficiency" may be used to state the value. Design is that the position where software development promotes efficiency. Design gives us software representations that will be tested for accuracy. Design is also the only way we can turn a vision of a consumer correctly into a finished product or program. Software design is a framework for all the successive steps of software engineering. We risk creating an unstable device without a powerful design – one that may be difficult to test, one whose consistency can't be tested until the final point.

Progressive refinement of knowledge structure during design, perspective of the project management. Technically, architecture consists of four practices – architectural design, device design, interface design, and procedural design.



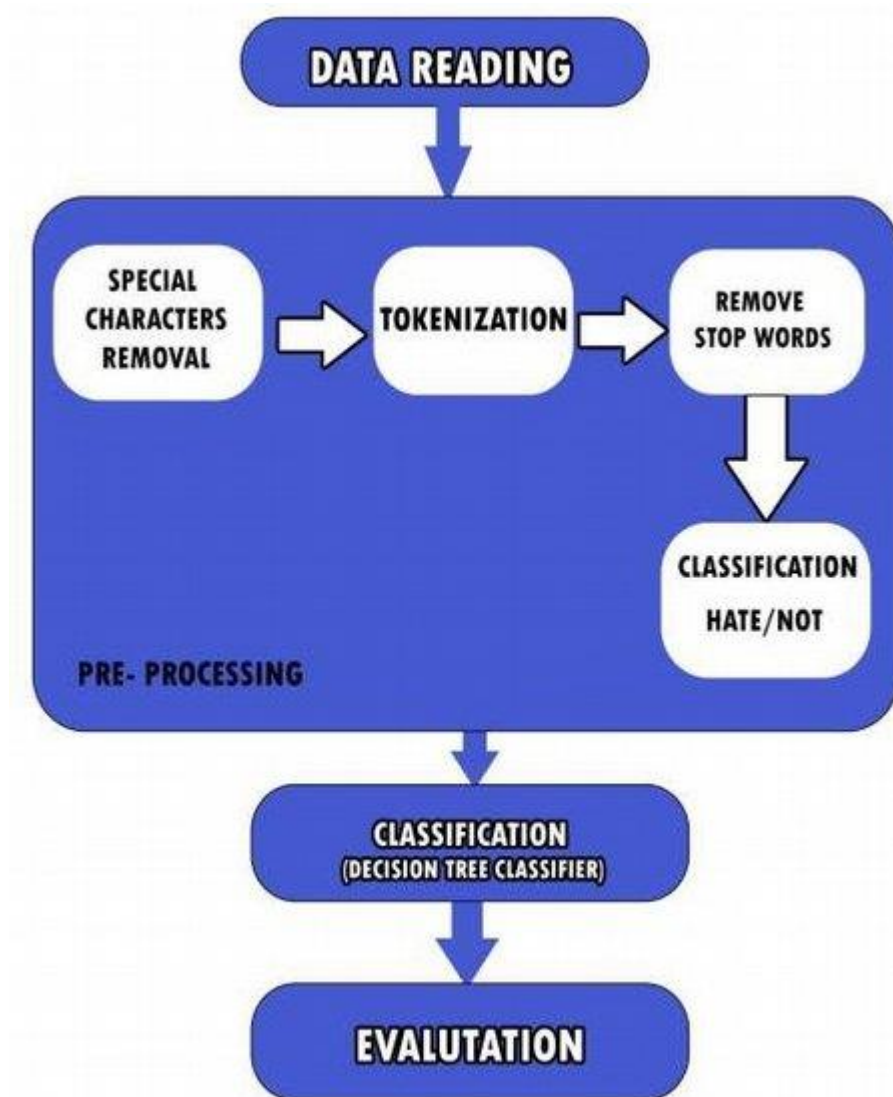


Figure 8: Architecture

4.2 MODULE DESCRIPTION

READING THE DATA

The data set in the form of .tsv is translated to the format of .csv. In order to perform some csv file operations, it is important to convert it to a data frame. To do this, a built-in method in python called 'read csv' is available. In a very library named pandas, this method has to be imported into our IDE.

PRE-PROCESSING

Pre-processing refers to the transformations that were applied before feeding on our data. In order to obtain better outcomes from the implemented model in Machine Learning projects, the data format must be very accurate. Every specified Machine Learning model requires information in a very specific format, for example, the Random Forest algorithm doesn't accept null values, so null values must be handled from the first raw data set to execute random forest algorithms. Another thing is that data set should be structured in such a simple way that quite a similar Machine Learning and Deep Learning algorithm is implemented in one set of data, and the best is chosen from them.

Steps Involved in Pre-processing:

There are 4 steps involved in Pre-processing:

- Special Characters removal
- Tokenization
- Stop word Removal
- Hate/Not Classification.

SPECIAL CHARCTERS REMOVAL:

The foremost process in Pre-processing stage is Special Characters removal. so as to form the given data ease for algorithm to spot each token and process the string this removal of special characters plays a significant role. during this stage we used Regular expressions to get rid of any non-alphanumeric characters.

re.sub (regex, string_to_replace_with, original_string)

TOKENIZATION:

Word tokenization is that the process of splitting an outsized sample of text into words. this can be a requirement in language processing tasks where each word must be captured and subjected to further analysis like classifying.

The language kit (NLTK) may be a library won't to achieve tokenization. A sentence or data are often split into words using the method **word_tokenize()**.

INPUT:

#Dhonikeepstheglove | WATCH: Sports Minister Kiren Rijiju issues statement backing MS Dhoni over 'Balidaan Badge', tells BCCI to take up the matter with ICC and keep government in the know as nation's pride is involved.

OUTPUT:

[dhonikeepstheglove, watch, sports, minister, kiren, rijiju, issues, statement, backing, msdhoni., balidaan badge, tells, bcci, take, matter, icc, keep, government, know, nations, pride, involved]

STOPWORDS REMOVAL:

Out[24]:
Stopwords are the words where a sentence doesn't bring much attention. They'll be ignored safely without losing the essence of the sentence. For example, sounds like "the," "with," "have," "is," "was".

There's no generic list of stop words, thus we took most of the common words in the sentences and matched our tokenized strings to those stop words list.

Given Input:

#Dhonikeepstheglove | WATCH: Sports Minister Kiren Rijiju issues statement backing MS Dhoni over 'Balidaan Badge', tells BCCI to take up the matter with ICC and keep government in the know as nation's pride is involved.

Output:

[dhonikeepstheglove, watch, sports, minister, kiren, rijiju, issues, statement, backing, msdhoni., balidaan badge, tells, bcci, take, matter, icc, keep, government, know, nations, pride, involved]

HATE/NOT CLASSIFICATION

Since our data set is in English, there's no in-built algorithm to feed the tokenized strings to algorithm to search out whether the given string is hating or not hence we took two .txt files during which one txt file consists of negative words list, and also the other positive words list. Each file consists of around 2000 words to check against the tokenized words. Hence every sentence within the data set which is tokenized is compared with negative words file and count is incremented and also compared against positive words and count is taken. These both the negative words count and positive words count is compared for polarity check

i.e., if the negative count for a sentence is larger than/greater than adequate to positive the polarity is going to be set to 1 else

Out[24]:

	neg	pos	polar
0	0	2	0
1	0	0	1
2	3	1	1
3	1	3	0
4	1	1	1
5	0	0	1
6	0	0	1
7	0	1	0
8	0	0	1
9	0	0	1
10	0	1	0
11	0	0	1
12	4	1	1
13	1	2	0
14	1	1	1
15	2	1	1
16	0	0	1
17	1	0	1
18	0	0	1
19	0	0	1
20	0	0	1

Figure 9: Determination of polarity scores

CLASSIFICATION

Classification is a matter of defining the group or groups (sub-populations) a new observation belongs to, on the basis of a training set of data with observations (or instances) whose membership in the category is understood. Classifier efficiency is highly dependent on the data properties to be categorized. No single classifier works best on any given question. Various empirical tests are performed to check the performance of the classifier and to search the information characteristics which determine the performance of the classifier. For a given question, however, determining an appropriate classifier is often more an art than a science. Precision and recall indicators are common metrics used to determine an organization's quality.

There are a number of classification techniques available, such as Decision Tree, Support Vector Machine (SVM), logical regression, Naive Bayes, we used the Tree classifier option to encourage better results.

DECISION TREE ALGORITHM:

Decision Tree Classifier could also be a basic classification method which is widely utilized. Decision trees all were seen as one of the its most commonly applied algorithms in machine learning, mostly for classification but more often in regression cases. Decision Tree Classifier asks a series of carefully constructed issues regarding qualities of test records. A add-up questions are asked upon having a comment before a decision can be made about name of the album group. Selecting spatial splits significantly impacts a tree's accuracy. The Classification and Regression trees eligibility requirements were special.

Decision trees use several algorithms to generate a push for a node to be divided into two or maybe more sub-nodes. The formation of forum-nodes improves the uniformity of the sub-nodes that arise. In other letters, we're willing to say that with relevance the aim function improves the potency of the node. The selection tree divides the partitions into all obtainable variables in order to pick the split ending of most stratified sub-nodes. Furthermore, the form of target variables is provided by the algorithm choice.

Decision Tree algorithms consist mainly of data for training, raw data, a function for heuristic validation and an operate for the stop criteria. These systems use the recursion of the divide and conquer system to elicit downwards data from node of inspiration. In the end, the decision trees are the "gold-standard" for

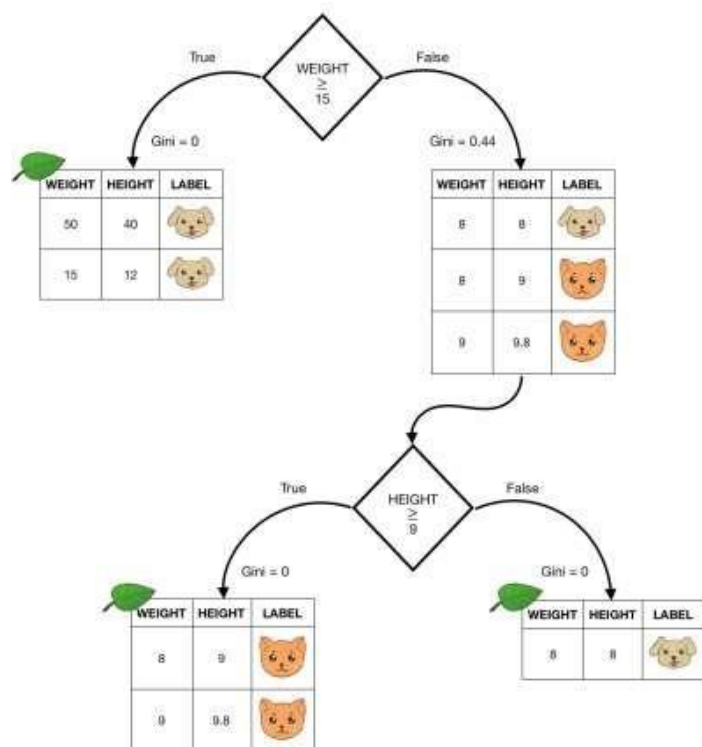
partition-based models. The decision trees it was first applied to the modeling of languages. One node was that the position to begin is preceded by binary queries which are asked as a way of arbitrarily partitioning the space of history. Due to the partitioning of the space, "leaves" are created and the training data is used to measure the probability of $P(w)$ for the next component. As the traversal continues the questioning becomes more insightful with the use of theoretical metrics of information.

WEIGHT (lbs)	HEIGHT (in)	LABEL
8	8	
50	40	
8	9	
15	12	
9	9.8	

Figure 10: A sample cat dataset

The concept of the Decision Tree while learning a dataset to identify a variable would be to split the data into shorter datasets centered on some kind of feature value until those target parameters fall under single umbrella.

Let's say that we want to create a decision tree to decide whether a pet is a cat, or a weight and height dependent dog. Such data sets may be divided focused, for example, on the values of one of the two characteristics: with a weight greater than 15 lbs, we are confident that the pet is a dog, at least judging on this very simple dataset. But if the weight is only around, I'll have two cats and one dog in



my subset, and I'll have to

divide the subset again till there is just single class left. In the other words, all my subsets are clean before they are.

Every tree starts with a root node, that is to say the first break. Without worrying too hard, we can probably come up with various types of root nodes to separate the dataset in the same number of steps, finding a solution.

Of course, it should search for the most effective way and we need to implement Gini in order to determine what this is: "the most widely used indicator of inequality." This inequality applies to the goal groups followed by a node in each subset.

For this reason, it can be calculated after each split, and depending on how the inequality changes after a node,

CONFUSION MATRIX

A confusion matrix could be a technique for summarizing the performance of a classification algorithm. the quantity of correct and incorrect predictions is summarized with count values and weakened by each class. this can be the key to the confusion matrix. Calculating a confusion matrix can provide you with a much better idea of what your classification model is getting right and what kinds of errors it's making.

The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

It gives your insight not only into the errors being made by your classifier but more importantly the categories of errors that are being made. it's this breakdown that overcomes the limitation of using classification accuracy alone.

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Figure 12: A model of confusion matrix

Here,

Class 1: Positive

Class 2: Negative

Definition of the Terms:

Positive (P): Observation is positive (for example: is an apple).

Negative (N): Observation is not positive (for example: is not an apple).

True Positive (TP): Observation is positive, and is predicted to be positive.

False Negative (FN): Observation is positive, but is predicted negative.

True Negative (TN): Observation is negative, and is predicted to be negative.

False Positive (FP): Observation is negative, but is predicted positive.

CLASSIFICATION RATE/ACCURACY

The accuracy of a machine learning classification algorithm is a method to live how often the algorithm classifies a datum correctly. Accuracy is that the number of correctly predicted data points out of all the info points. More formally, it's defined because the number of true positives and true negatives divided by the

metrics that use various ratios of true/false positives/negatives. Together, these metrics provide an in depth observe how the algorithm is classifying data points. Consider a classification algorithm that decides whether an email is spam or not. The algorithm is trained, and that we want to determine how well it performs on a collection of ten emails it's never seen before. Of the ten emails, six aren't spam and 4 are spam. The algorithm classifies three of the messages as spam, of which two are literally spam, and one isn't spam. within the table, truth positives (the emails that are correctly identified as spam) are coloured in green, truth negatives (the emails that are correctly identified as not spam) are coloured in blue, the false positives (the not spam emails that are incorrectly classified as spam) are coloured in red, and therefore the false negatives (the spam emails that are incorrectly identified as not spam) are coloured in orange. There are two true positives, five true negatives, two false negatives, and one false positive. Using the formula for accuracy, we get:

This algorithm has 70% accuracy classifying emails as spam or not.

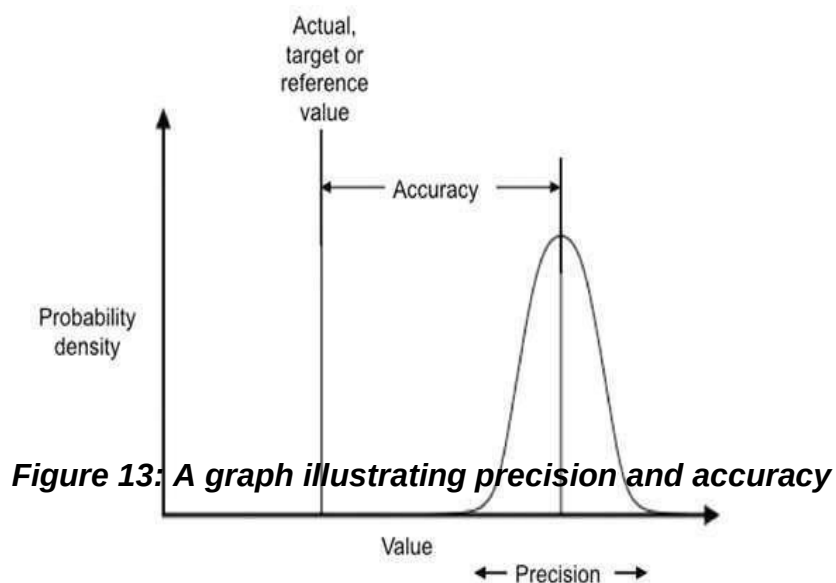


Figure 13: A graph illustrating precision and accuracy

$$Precision = \frac{truepositives}{truepositives + falsepositives}$$

Figure 14: Formulas for calculating precision, recall and accuracy

However, there are problems with accuracy. It assumes equal costs for both types of errors. A 99% accuracy may be excellent, good, mediocre, poor or terrible depending upon the matter.

PRECISION AND RECALL

In pattern recognition, knowledge acquisition and classification (machine learning), precision (also known as positive predictive value) was that the fraction of relevant incidents among all the obtained circumstances, while recall (also known as sensitivity) is that portion of the total number of sensitive contexts which were ultimately collected. Therefore, a comprehension and measure of significance are provided by both precision and recall.

PRECISION

Precision improves when there is a high price for false positives. And let's say the matter is about carcinoma detection. When we have a very low-precision model, so several patients are advised they need melanoma, which will have certain exacerbations. There's plenty of rigorous checking and tension involved. When false positives are too heavy, after being barraged with false reports, persons who track the outcome will learn to recognize these. To encourage accuracy, we divide the total amount of correctly identified good examples by total number of similar cases expected. Large precision implies a strong example (a small number of FP) classified as positive.

Figure 15: Formula for calculating the precision

RECALL

Remember helps when false negatives are priced high. A false negative has catastrophic effects. False execute and we all die. When false negatives are

prevalent, the thing you want to stop gets you hit. If you plan to disregard the vibration of a twig broken in an extremely dark wood, a false negative is and you get eaten by a cub. (A false positive is to stay sleepless all night in your tent in an excessively nervous state, keeping track of every noise in the woods, only to learn the morning after that those noises were made by a chipmunk. Not fun.) If you had a model that mistakenly makes nuclear weapons, you'd like to throw this out.

$$Recall = \frac{truepositives}{truepositives + falsenegatives}$$

Figure 16: A formula for calculating Recall

High recall, low precision: This implies that most of the positive examples are correctly recognized (low FN) but there are plenty of false positives.

Low recall, high precision: This shows that we miss plenty of positive examples (high FN) but those we predict as positive are indeed positive (low FP).

F -MEASURE: F1 is an aggregate indicator of the accuracy of a model that combines precision and recall, in this interesting way of adding and multiplying merely combines two items together just to create a separate bowl. That is, a good F1 score means you've got low false positives and low false negatives, so you recognise real threats correctly and you shouldn't be bothered by false alarms. Great consideration is given to an F1 score when it becomes 1 while the template is a complete failure and when 0. Since we have two measurements (Precision and Recall) it helps to own a measurement which portrays either. We quantify an F-measure that uses in situ mean of absolute value, since it more punishes the extreme values.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Figure 17: Formula for calculating the F-score

Perks of using a contradictory matrix. It reveals how frustrating every classification model was when making predictions. Confusion matrix not just to provides your intuition with your classifier's faults, it also provides you any kind of mistakes that are made. This summary allows you to solve the drawback of using

precision classification only. Each column of the matrix of confusion reflects the cases of that predicted class. Every row of the confusion matrix reflects the specific class of instances. It gives knowledge not only of the mistakes made by a classifier but also of the bad decisions made.

PROGRAMMING LANGUAGE

Python is a language of programming that is translated, of high complexity, and general use. Designed by Guido van Rossum and written in 1991. The programming ideology of Python stresses the readability of code with its impressive usage of great whitespace. The language structures and object-oriented methodology are aimed at helping programmers create simple, functional code for projects of small and large scale.

Python is written dynamically, and obtained from trash. It supports multiple programming concepts, including organized programming (especially procedural), object-oriented, and functional. Because of its comprehensive standard library, Python is often represented as "batteries included" language.

INTRODUCTION TO TECHNOLOGY

NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) can be a computational area in which computers interpret, comprehend and infer significance from human language in a rather knowledgeable and useful manner. Through utilizing NLP, developers can coordinate and arrange information to conduct tasks such as automated summing, encoding, identification of named persons, abstraction of associations, analysis of emotions, identification of expression, and standardization of subjects. NLP is used to interpret language, making it easier for computers to learn how people talk. This connection between person and machine enables real-world applications such as automated text summarization, emotion interpretation, subject extraction, identification of named persons, labeling of sections of expression, abstraction of associations, halting, and more. NLP is also used for text mining, MT, and the addressing of automated queries.

Open-Source Libraries:

Such libraries have real-world implementations for the algorithmic foundations of NLP. Algorithms includes a free endpoint API for some of such algorithms, without building up or running servers and storage at all.

- **Apache OpenNLP:** A machine learning toolkit of tokenizers, fragmentation of words, part-of-speech labelling, specified extracting of items, chunking, sorting, coreference resolution, and far more.
- **Natural Language Toolkit (NLTK):** A Python framework with plugin for word processing, sorting, tokenization, halting, labeling, filtering, and more.
- **Stanford NLP:** A series of NLP tools to include part-speech marking, the defined object recognizer, coreference negotiation framework, sentiment analysis and more
- **MALLET:** A Java package that has Latent Dirichlet Allocation, sorting of records, clustering, simulation of subjects, extraction of information, and more.
- To accomplish the language processing and us the Sentimental Analysis we'd like an open-source library named language **NATURAL LANGUAGE TOOL KIT(NLTK).**

REASONS FOR CHOOSING NLTK OVER OTHER OPEN SOURCE LIBRARIES:

- **NLTK vs Mallet:** MALLET is java-based package for machine learning projects that are mainly deployed in industrial purposes. MALLET is more geared toward professional purpose and also highly complex to use whereas the NLTK is more user-friendly and also it contains more documentation compared to MALLET.
- **NLTK vs STANFORD NLP:** STANFORD NLP is additionally a python-based library in similar with NLTK. The speciality of STANFORD NLP is to a part of speech tagger. But the range of libraries provided by the NLTK lags in STANFORD NLP. STANFORD NLP has only one window such it can't be accessed by the third-party site. Once if this classifier able to operate it should be able to interact with numerous sites. With an amateur

knowledge we've got decided that to change with NLTK in situ of STANFORD NLP.

- **NLTK vs APACHE open NLP:** during this instance also, the most problem is primarily the programming language used. APACHE open NLP uses the java as programming language and it had been released sooner than NLTK such a large amount of of the standard users feel their comfortable levels in OPEN NLP than NLTK. But as NLTK is primarily focuses than text analysis kind of like OPEN NLP but the benefit of accessing the built-in libraries and also the inbuilt corpus is add-on feature for language Tool kit.

TOOLS USED

NATURAL LANGUAGE TOOLKIT

NLTK may be a pioneering tool for developing Python programs with human language data to work out. It offers quick-to-use frameworks for more than 50 organizational and lexical tools such as WordNet, along with a set of text processing libraries for grouping, tokenization, halting, labeling, filtering, and semance reasoning, wrappers for NLP repositories of institutional intensity, and a vibrant debate board.

The language toolkit, or more generally NLTK, may be a collection of libraries and systems built within the Python programming language for functional and mathematical language processing (NLP) for English. Steven Bird and Edward Loper had created it inside the University of Pennsylvania's Department of Computer and Information Technology. NLTK provides experimental details and interactive proofs. It is in the center of a book which explains the fundamental concepts behind the toolkit-supported language processing tasks, and a cookbook.

NLTK is meant to help NLP or similarly associated study and training, including analytical linguistics, academic science, computation, knowledge processing, and machine learning. NLTK was widely used as a teaching method, as a private analysis device, and as a forum for the machining and development of test

structures. NLTK provides advanced features for grouping, tokenization, trailing, labeling, sorting, and semantic thinking.

Modules used in building model:

- **corpus ()**: Corpus can well be a series of written documents. You've got several corpora in NLTK, including Gutenberg Corpus, then on Internet and Chat Script.

Submodule **stopwords()**: A stop word might well be a widely encountered term (such as "the," "a," "an," "in") that has been configured to avoid a groundwork machine, both when indexing search inputs and when recovering them regardless of the effects of a foundations question.

- **tokenize ()** : Tokenisation is the method by which vast volumes of text are separated into smaller pieces called tokens. To break a sentence into words we use the `word_tokenize()` process. In machine learning systems, the performance of word tokenisation may also be translated to Data Frame for improved text comprehension. As an entry for more text cleaning moves such as punctuation elimination, numerical character elimination or trailing, it is received visiting. Machine learning models need to practice numerical data and render a forecast. Word tokenization to numerical translation is a very necessary aspect of the text (string).

- **sklearn()**: Scikit-learn (also named sklearn) might well be a sample library for the Python language training computer program. It supports numerous grouping, regression, and clustering algorithms which include supporting vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is intended to modularize with the NumPy and SciPy Python mathematical and science libraries.

Submodules used:

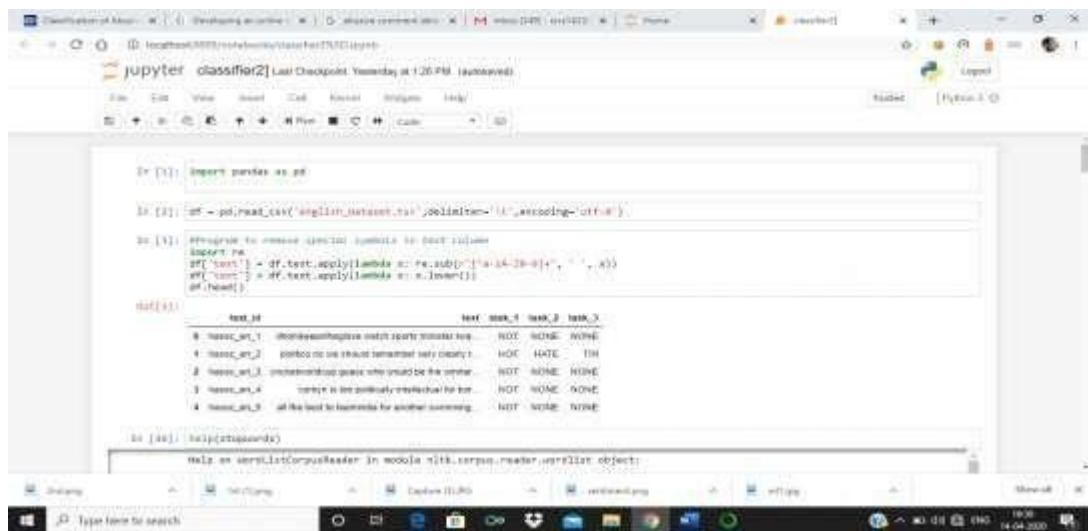
LabelEncoder: Encodes target labels with value between 0 and `n_classes-1`.

Tree: A Decision Tree Classifier

Train_test_split(): Split arrays or matrices into random train and test subsets.

Chapter – 5, Results and Discussion, Performance Analysis

5.1 SNAPSHOTS OF THE PROJECT



The screenshot shows a Jupyter notebook interface with the following code and output:

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('english_dataset.csv', delimiter='|', encoding='utf-8')

In [3]: #Prepwork to remove special symbols so that it is clean
import re
df['text'] = df.text.apply(lambda x: re.sub('[^a-zA-Z0-9]+', '', x))
df['text'] = df.text.apply(lambda x: x.lower())
df.head()
```

Out[3]:

	text_id	text	sent_1	sent_2	sent_3
0	tweet_0_1	donkeyandthepigeon don't report similar too	NOT	NONE	NONE
1	tweet_0_2	perhaps no one should remember very clearly.	NOT	HATE	TH
2	tweet_0_3	indeedandthepigeon who should be the winner.	NOT	NONE	NONE
3	tweet_0_4	maybe in the probablyindeedbut for too	NOT	NONE	NONE
4	tweet_0_5	all the best to everyone for another journey.	NOT	NONE	NONE

In [4]: help(str.strip)

help on str.strip in module re: re.compile object:

Figure 48: Snapshot-1 of Model in Anaconda Jupyter notebook



Figure 50: Snapshot-3 of Model in Anaconda Jupyter notebook

The screenshot shows a Jupyter notebook with a Python script and its output. The script defines a function `def model(x):` which uses `sklearn.metrics.classification_report` to evaluate a model. The output is a table showing the results of the model evaluation.

```
def model(x):
    y_pred = model.predict(x)
    y_true = y
    report = classification_report(y_true, y_pred)
    return report
```

	prec	recall	f1	support
0	0.95	0.95	0.95	1000
1	0.95	0.95	0.95	1000
2	0.95	0.95	0.95	1000
3	0.95	0.95	0.95	1000
4	0.95	0.95	0.95	1000
5	0.95	0.95	0.95	1000
6	0.95	0.95	0.95	1000
7	0.95	0.95	0.95	1000
8	0.95	0.95	0.95	1000
9	0.95	0.95	0.95	1000
10	0.95	0.95	0.95	1000
11	0.95	0.95	0.95	1000
12	0.95	0.95	0.95	1000
13	0.95	0.95	0.95	1000
14	0.95	0.95	0.95	1000
15	0.95	0.95	0.95	1000
16	0.95	0.95	0.95	1000
17	0.95	0.95	0.95	1000
18	0.95	0.95	0.95	1000
19	0.95	0.95	0.95	1000
20	0.95	0.95	0.95	1000
21	0.95	0.95	0.95	1000
22	0.95	0.95	0.95	1000
23	0.95	0.95	0.95	1000
24	0.95	0.95	0.95	1000
25	0.95	0.95	0.95	1000
26	0.95	0.95	0.95	1000
27	0.95	0.95	0.95	1000
28	0.95	0.95	0.95	1000
29	0.95	0.95	0.95	1000
30	0.95	0.95	0.95	1000
31	0.95	0.95	0.95	1000
32	0.95	0.95	0.95	1000
33	0.95	0.95	0.95	1000
34	0.95	0.95	0.95	1000
35	0.95	0.95	0.95	1000
36	0.95	0.95	0.95	1000
37	0.95	0.95	0.95	1000
38	0.95	0.95	0.95	1000
39	0.95	0.95	0.95	1000
40	0.95	0.95	0.95	1000
41	0.95	0.95	0.95	1000
42	0.95	0.95	0.95	1000
43	0.95	0.95	0.95	1000
44	0.95	0.95	0.95	1000
45	0.95	0.95	0.95	1000
46	0.95	0.95	0.95	1000
47	0.95	0.95	0.95	1000
48	0.95	0.95	0.95	1000
49	0.95	0.95	0.95	1000
50	0.95	0.95	0.95	1000
51	0.95	0.95	0.95	1000
52	0.95	0.95	0.95	1000
53	0.95	0.95	0.95	1000
54	0.95	0.95	0.95	1000
55	0.95	0.95	0.95	1000
56	0.95	0.95	0.95	1000
57	0.95	0.95	0.95	1000
58	0.95	0.95	0.95	1000
59	0.95	0.95	0.95	1000
60	0.95	0.95	0.95	1000
61	0.95	0.95	0.95	1000
62	0.95	0.95	0.95	1000
63	0.95	0.95	0.95	1000
64	0.95	0.95	0.95	1000
65	0.95	0.95	0.95	1000
66	0.95	0.95	0.95	1000
67	0.95	0.95	0.95	1000
68	0.95	0.95	0.95	1000
69	0.95	0.95	0.95	1000
70	0.95	0.95	0.95	1000
71	0.95	0.95	0.95	1000
72	0.95	0.95	0.95	1000
73	0.95	0.95	0.95	1000
74	0.95	0.95	0.95	1000
75	0.95	0.95	0.95	1000
76	0.95	0.95	0.95	1000
77	0.95	0.95	0.95	1000
78	0.95	0.95	0.95	1000
79	0.95	0.95	0.95	1000
80	0.95	0.95	0.95	1000
81	0.95	0.95	0.95	1000
82	0.95	0.95	0.95	1000
83	0.95	0.95	0.95	1000
84	0.95	0.95	0.95	1000
85	0.95	0.95	0.95	1000
86	0.95	0.95	0.95	1000
87	0.95	0.95	0.95	1000
88	0.95	0.95	0.95	1000
89	0.95	0.95	0.95	1000
90	0.95	0.95	0.95	1000
91	0.95	0.95	0.95	1000
92	0.95	0.95	0.95	1000
93	0.95	0.95	0.95	1000
94	0.95	0.95	0.95	1000
95	0.95	0.95	0.95	1000
96	0.95	0.95	0.95	1000
97	0.95	0.95	0.95	1000
98	0.95	0.95	0.95	1000
99	0.95	0.95	0.95	1000

Figure 51: Snapshot-4 of Model in Anaconda Jupyter notebook

5.2 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM

- Text collection (dataset elicitation)
- Text Pre-processing
- Text Transformation
- Feature Selection
- Pattern Discovery
- Interpretation and Evaluation

Actually, this project deals with the analysis of comment and prevention the hate, offensive and profane words. According to the HASOC data set collected the we need to treat the text in a proper manner in order to obtain optimal results. Actually, here the data will be provided by the user but here as there will very less and secure access to the social media sites, we have taken predefined dataset provided at FIRE-2019 i.e., the Indo- European Languages. So, we have done analysis on two languages but the tests cases and the other illustrational activities are represented in only English language. Despite the Project covers English and Hindi but the stopwords sub-module is not predefined for Hindi language. So, the main challenge here is that we can remove the stopwords and also the special characters ain Hindi are also not predefined. So, the illustrations and further proceedings are carried out only in English Language dataset.

Text Collection:

Here the text is collected using the twitter dataset and the we have an self sufficient number of comments or tweets to highlight the need to analyse and also detect the bad words in the given twitter dataset. The Twitter Data set is taken

from the recent events and real-time tweets at HASOC workshop at FIRE-2019. The Text which is to be pre-processed is collected and they are made into text file. File can be of any format i.e., the csv or txt file. But the to our convenience and the depending on the system requirements we are considering a text file i.e., in the format of txt file.



Figure 54: A Raw Data set of Text from twitter

Text Pre-processing

In the pre-processing of the text for the raw data we have many challenges of clearing the special characters and also making it into a set of words so that they can be used for further analysis and also pattern recognition can be achieved optimally, so in this order of events we have some peculiar tasks but they prove to be important in the further proceedings of building the model. Here in comments, we tend to use the hastags(#) and also underscores (_) and many other special characters to represent the importance of the post and also to highlight the events and posting times. Also, we use different cases of letters to maintain grammatical balance. This makes a bit difficult to analyse. As python is case sensitive so in order to make it convenient, we have to convert all the characters into single case either into lowercase. As universally used and more evident case is lower case, we have used and converted into the lower-case letters.

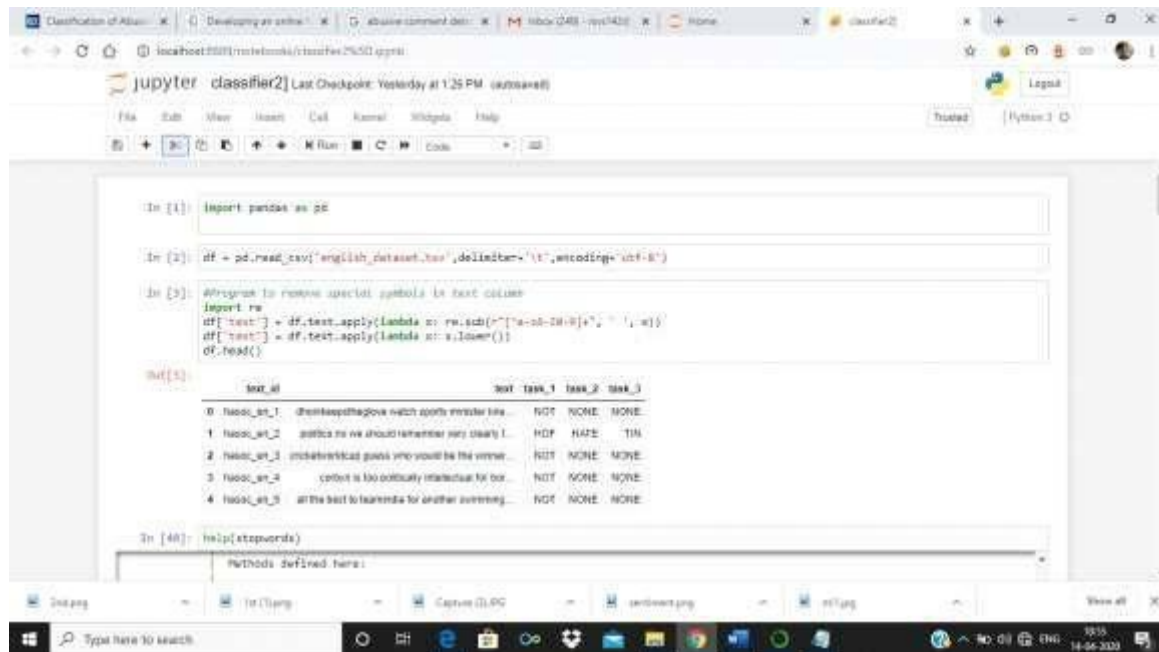


Figure 55: Data Preprocessing

Text Transformation

As the first step is carried out there is a need to make it more organised for the further analysis. So, the need to make the words into groups has arisen. But in the further analysis we need to calculate the polarity scores for the group of words. It became difficult to calculate the polarity scores and also carry out sentiment analysis for numerous sets of words. So, it is hardly needed to remove the unwanted words that don't affect the meaning of the sentence. So, these words are predefined listed by the NLP toolkit NLTK and named as stopwords. This stopwords sub-module is present in corpus module

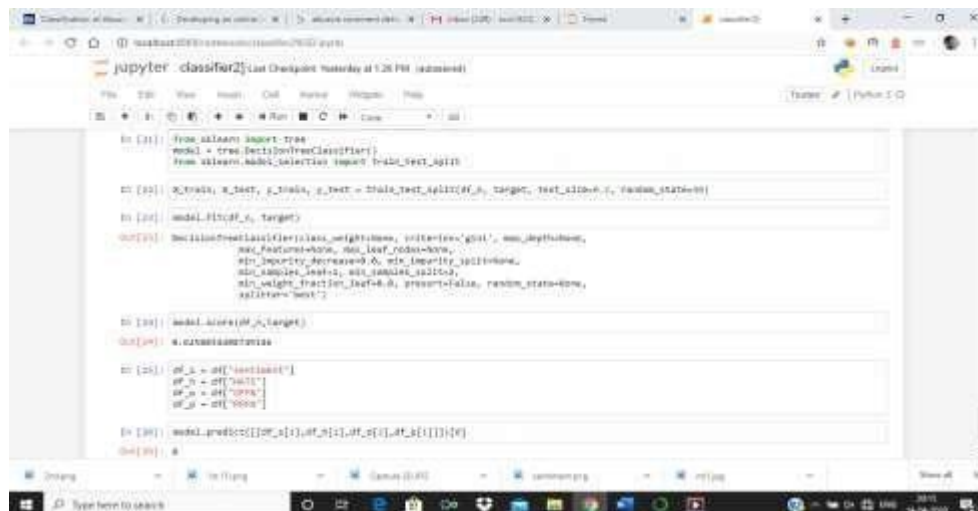
Also, after removing the stopwords we have to make some pattern son that it can be effective recognition of the text. In this model we are transforming the processed text into lists. Lists can easily and optimally handle in python.

In this step of process basic calculations and the other simple methodologies are used to achieve the optimality in the system. Firstly, by using the vaderSentimentAnalysis module we try to analyse the sentimental intensity and we try to calculate the polarity scores of the corresponding lists. If the score is more than predicted score (0.05) then it will be returned with 1 or else with 0.

Also, some simple statistics and calculations have been calculated to count the number of words according to the category like hate, offensive and profane.

Data Mining

Here the classification of data takes place and we try to classify the data and put it into a decision tree and analyse it based on the results acquired on the dataset polarity scores. In this process of training and verification of the module we used the decision tree algorithm. This decision tree algorithm is present in the sklearn machine learning module which is predefined in the Natural Language Tool Kit (NLTK).



```
from sklearn import tree
model = tree.DecisionTreeClassifier()
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, target, test_size=0.3, random_state=0)

model.fit(X_train, y_train)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                        max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                        splitter='best')

model.score(X_test, y_test)
Out[26]: 0.8333333333333333

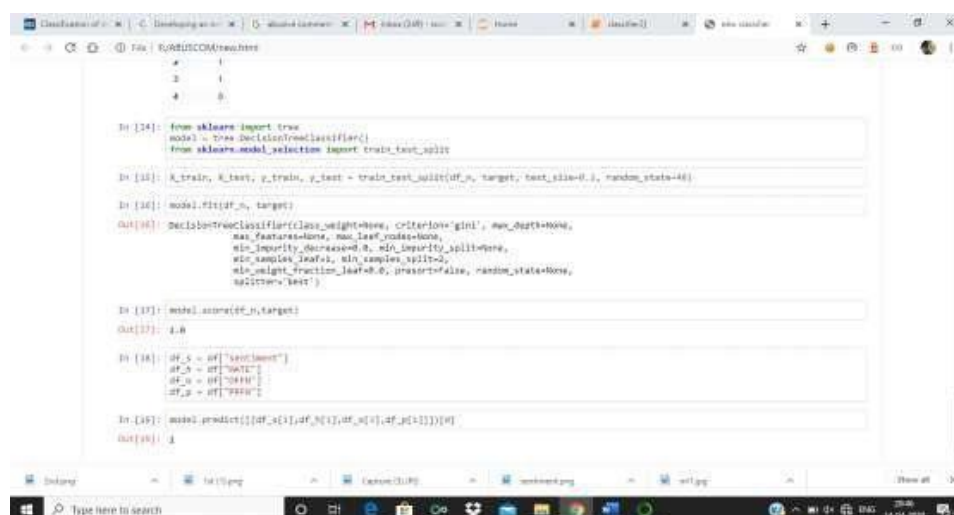
df_s = df["sentiment"]
df_n = df["neg"]
df_p = df["pos"]

model.predict([df_s[1],df_n[1],df_p[1],df_s[1]])[0]
Out[26]: 0
```

Figure 59: Data Mining

Interpretation and Evaluation:

In this phase the predicted scores and other evaluation matrices are calculated to maintain and in efficient building of model.



```
from sklearn import tree
model = tree.DecisionTreeClassifier()
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, target, test_size=0.3, random_state=0)

model.fit(X_train, y_train)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                        max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                        splitter='best')

model.score(X_test, y_test)
Out[27]: 0.8

df_s = df["sentiment"]
df_n = df["neg"]
df_p = df["pos"]

model.predict([df_s[1],df_n[1],df_p[1],df_s[1]])[0]
Out[28]: 1
```

Figure 60: Interpretation and Evaluation of Model

- **Classifier.ipynb:** It is the heart and soul of the project as the implementation code is basically in this classifier.ipynb. This python notebook contains each and every line of implementation and modification in this file reflects the usage of the classifier and also its respects.
- **english_dataset.tsv:** It is the tsv file in which the twitter data is stored. The received tweets are stored in this file, for analysis.
- **hate.txt:** It is a text file for storing the hate words after/ in the classification process.
- **List.txt:** It is a text file for storing the profane words after/ in the classification process.
- **Offensive.txt:** It is a text file for storing the offensive words after/ in the classification process.

Chapter – 6, Summary and Conclusions

6.1 CONCLUSION

The research aimed to assess the efficiency of offensive language identification in terms of specificity, accuracy and recall. In this, we contrasted specific supervised Decision Tree machine learning algorithms to recognise offensive tweets in twitter. It offers us a fascinating look into how hate-mongers utilize trends in terms of how they convey hatred, sexism and misinformation. The experimental findings indicate that for the aggressive tweets analysis the classifiers produced stronger outcomes with the decision tree solution providing more than 90 percent precision and outpacing other algorithms.

6.2 FUTURE SCOPE

For future projects we proposed using NB as a reference result of word unigram and the combination of word n-gram. Offensive rhetoric is usually a component of hate speech. And the researcher who is interested in hate speech detection research could use the dataset.

We intend to improve the recall by adding various algorithms for future research, and extend the test collection to include tweets from several regions that include specific dialects and cultures. In comparison, the research in this paper centred on the detection of inflammatory messages, and we intend to extend it to include expression of hatred. Furthermore, in addition to unigrams and bigrams, we expect to research various stages of morphological and syntactic interpretation and use character n-grams to cope with the rich morphology. Ideally, morphological analysis will result in improved recall.

CODING:

```
Import pandas as pd
Pd=pd.read_csv('english_dataset',delimiter='/',encoding='utf-8')
Help(stopwords)
Pip install NLTK
Import nltk
Nltk.download('stopwords')
#Program to remove stopwords
#Program to tokenize the words
from nltk.corpus import stopwords
From nltk.tokenize import word_tokenize
Def stopwrds(x):
Stop_words =set(stopwords.words('english'))
Word_tokens=word_tokenize(x)
Filtered_sentence=[w for w in word_tokens if not w in stop _words]
Filtered_sentence=[]
For w in word_tokens
If w not in stop _words:
Filtered_ sentence.append(w)
Return filtered_sentence
Df['text_split']=df.text.apply(stopwrds)
Df.head()
#Function to analyze the sentiment
From vader sentiment.vadersentiment import sentimentintensity
Sid_obj =sentiment intensity Analyzer
Def senti(x)
Sentiment_dict =sid_obj.polarity_score(x)
```

```

If sentiment_dict['compound']>=0.05:
Return 1
Elif sentiment_dict['compound'] <=-0.05:
Return 0
Else:
Return 1
Df['sentiment']=df.text.apply(senti)
Df.head()
Import codes
Infile = codecs.open("list.txt","r", utf-8")
Num=infile.read()
Num=num.replace('\n')
Num=num.replace(' ','')
Num=num.replace('","')
Profane=num.split(',')
Len(profane)
Filehate=open("hate.txt","r")
Num =filehate.read()
Num =num.split('\n')
B=set ()
Hate= []
For x in num:
If x not in b:
Hate.append(x)
b.add(x)
Len(hate)
Fileoff=open ("offensive.txt","r")
Num =fileoff.read()
Num =num.split('/n')
B=set ()
Offensive = []
For x in num:
If x not in b:
Offensive. append(x)
b.add(x)
Len (offensive)

```

```

Def count(x,y):
Cnt =0;
For z in x:
If z in y:
If senti(z)==0:
Cnt+=1
Return cnt
Def counthat(x,y):
Cnt =0;
For z in x:
If z in y:
Cnt+=1
Return cnt
Df['hate']=df.text_split.apply(lambda x:counthat(x,hate))
Df['offn']=df.text_split.apply(lambda x:count(x,offensive))
Df['prfn']=df.text_split.apply(lambda x:counthat(x,profane))
Df.head()
From sklearn.preprocessing import label encoder
Le_task_1=label encoder ()
Df['task_1_n']=le_task_1.fit_transform(df['task_1'])
Df.head()
Df_n
Df.drop('text_id','text_split','task_2','task_3','task_1_n', axis=;column=ns')
Df_n.head()
Target.head()
#Function to build tree
From sklearn import tree
Model =tree.decisiontreeclassifier()
From sklean.model_selection import train_test_split
X_train,x_test,y_train,y_test  =  train_test_split(df_n,target,test_size=0.2,
random_state=40)
Model.fit(df_n,target)
Model.score(df_n,target)
Df_s=df["sentiment"]
Df_h=df["HATE"]
DF_o=df["OFFN"]

```

Df-p=df[“PRFN”]

Model.predict([[df_s[1],df_h[1],df_o[3],df_p[1]]]) [0]

REFERENCES (sort chesuko alphabetical order)

[1] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 10(4):215–230.

[2] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web, pages 29–30, New York, NY, USA. ACM.

[3] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Detection of cyberbullying incidents on the instagram social network. CoRR, abs/1503.03909.

[4] Sara Owsley Sood, Elizabeth F. Churchill, and Judd Antin. 2012b. Automatic identification of personal insults on social news sites. J. Am. Soc. Inf. Sci. Technol., 63(2):270–285, February.

[5] Leandro Araujo Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In Proceedings of the Tenth International Conference on Web and social media, pages 687–690, Cologne, Germany.

[6] Peter Burnap and Matthew Leighton Williams. Hate speech, machine classification and statistical modelling of information flows on Twitter. In Internet, Policy and Politics, 2014.

[7] Peggy McIntosh, 2003. Understanding prejudice and discrimination., chapter White privilege: Unpacking the invisible knapsack, pages 191–196. McGrawHill.

[8] Pete Burnap and Matthew L. Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, 5(1):1–15.