**Mini Project Report on**



**Detecting Hate Speech**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Kshitij Koul**  **2018895**

***Under the Mentorship of***

**Dr. Arun Chauhan**

**Associate Professor, Dept. of CSE**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

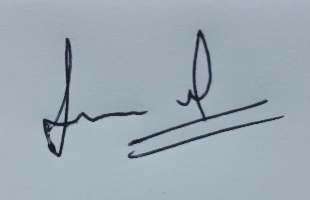
**July-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Detecting Hate Speech”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Arun Chauhan, Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Kshitij Koul 2018895



**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-4** |
| Chapter 3 | Methodology | **5-9** |
| Chapter 4 | Result and Discussion | **10-11** |
| Chapter 5 | Conclusion and Future Work | **12** |
|  | References | **13** |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

**MOTIVATION OF WORK:**

Any speech that disparages a group of people because of their race, religion, nationality, origin, sexual orientation, and or gender identity is considered as **Hate Speech**. Hate Speech Detection is basically the task of detecting if any kind of communication like audio, verbal, text, etc. contains hatred and or encourages violence towards a group of people or any particular person. With the use of Natural Language Processing (NLP), hate speech detectors can automatically examine or scan text and then assign a set of pre-defined tags based on the behaviour of content. Hate Speech Detection is becoming an increasingly important part of businesses as it allows to easily get some perception and understanding of people regarding their product and it also helps to analyze the percentage of people talking negatively about the organization’s product. Some of the most common examples and use cases for hate speech detection include the following: can be used mainly in social media platforms to detect and remove hate speech from user’s post, can be used to monitor discussions in online forums and communities, can be used in news sites and comment sections to filter hate comments and promote respectful discourse, it can be used in educational platforms to flag any kind of derogatory language, also it can be used in e-commerce platforms for monitoring product reviews and people’s view. Because of the complex and tangled nature of speech, analyzing, recognizing, organizing, and categorizing through given text data is hard and time-consuming, so most companies fail to use it to its full potential. At the same time identifying a kind of speech and rectifying it if it is wrongly categorized is a significant concern for any business organization, MNC’s, etc. so here comes machine learning in action which can help in doing so.

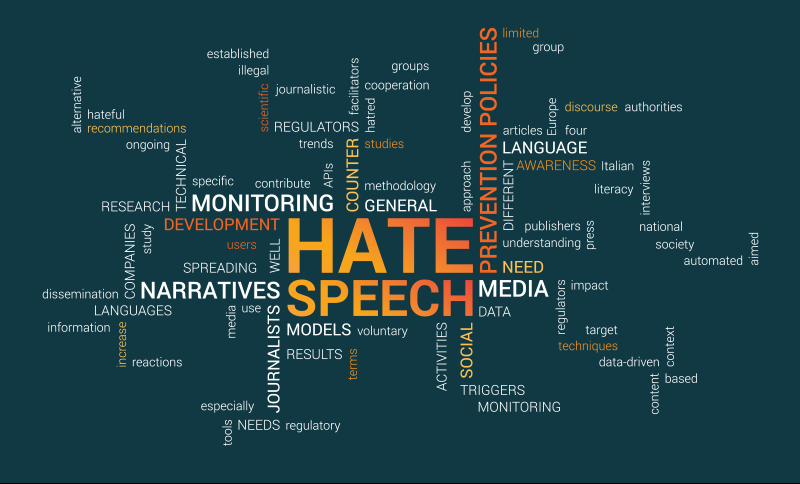
In order to predict that which category of speech the given input belongs to, one can use the concept of machine learning algorithms like I have used in my project and it has shown promising results in the speech categorization. The basic objective of this project is given as follows: -

1. Predict the nature of speech (whether offensive or not) to which the given data/text belongs to.
2. Effectively provide a model for accurate hate speech detection.
3. By detecting hate speech, we can protect individuals and groups from harmful and offensive language which may lead to psychological harm.
4. By detecting hate speech, we can maintain safe and inclusive environment on online platforms and promote respectful communication.

**1.2 Problem Statement**

The problem statement of the mini-project can be defined as:

The primary goal of the hate speech detection model is to examine the input which is taken as new data and then conclude on what kind of speech (whether offensive or not) does the given input belongs to.



**Fig. 1.1 Hate Speech Detection functionalities [1]**

**Chapter 2**

**Literature Survey**

Lot of work has formerly been done for the detection of hate speech effectively and in this section some of these works have been discussed in detail.

**2.1 Multi-modal Hate Speech Detection using Machine Learning [2]:**

With persistent development of web clients and media substance, it is exceptionally difficult to track down hateful speech in sound and video. Changing over video or audio into text does not identify hate speech precisely as humans sometimes use pleasant words as hate speech in a derisive way and also uses distinctive voice tone or shows different activity in the video than text. The state-of-the-art hate speech detection models were created on a single speech methodology. This research paper provides an integrated approach to identify hate speech from contents by utilizing video, audio and speech by drawing out feature images, feature values extracted from audio, text and used machine learning and natural language processing to identify hate speech.

**2.2 Hate Speech Detection using Machine Learning [3]:**

A plethora of methods have as of now been developed for the mechanization of hate speech detection online. There are two components to this process: recognizing the qualities that these terms utilize to target a certain bunch and categorizing textual material as hate or not hate speech. Due to time constraints, research efforts are started on the latter issue in this project. For this reason, identifying hate speech is a more challenging endeavor, as the research of the language used in typical datasets discloses that hate speech lacks distinctive and discriminatory characteristics. Deep neural network topologies are extremely valuable for capturing the meaning of hate speech and are hence proposed as feature extractors. In this research paper, text information from social media platforms such as Twitter are utilized to test the adequacy of these procedures, and they uncover a 6-percentage point enhancement in macro-average F1 or a 9 percent advancement for content that has been labeled as repulsive, respectively.

**2.3 Hate Speech Detection Using Decision Tree Algorithm [4]:**

The evolution of the web and social media has led to individuals having access to a plethora of platforms to freely express their thoughts and opinions on numerous topics. However, this freedom of expression is misused to ignite hatred against groups of people or individuals grounded on race, religion, gender, etc. Therefore, in order to address this developing problem on social media sites, recent studies have utilized various feature engineering techniques and machine learning algorithms to automatically identify hate speech posts on diverse datasets. Rapid changes in machine learning have captivated researchers looking for implementing solutions to the problem of hate speech. Currently, we are making use of decision tree algorithm technique to detect hate speech using text data.

**On the Next page we have discussed the Methodology of the Project.**

**Chapter 3**

**Methodology**

**3.1 PROPOSED SYSTEM:**

1. In this system, Decision Tree *(machine learning)* [3.2] algorithm is used, more specifically the supervised learning algorithm, can be used for classification tasks.

2. The model is trained using well “labelled” training data, and the system is capable of estimating the result on the basis of that data. Here, the labelled data mainly refers that some input data is formerly tagged with the correct output.

3. After training the model, the system can conclude on what kind of speech does the input text belongs to.

**3.2 MACHINE LEARNING:**

Machine Learning is said to be as a subdivision of artificial intelligence that is significantly concerned with the development of algorithms which allow a computer to learn from the data and experiences of past events on their own. The term machine learning was first devised by Arthur Samuel in 1959. Machine Learning is an emerging technology which uses numerous algorithms for constructing mathematical models and making predictions using historical data/information. Moreover, Machine learning can be classified into three types: Supervised learning, Unsupervised learning and Reinforcement learning.

Firstly, we will start with the essential packages and libraries which are required for this project and they are as follows: -

**3.3 NUMPY:**

NumPy is a library for the Python programming language, which adds a support for large, multi-dimensional arrays and matrices along with a huge collection of high level and comprehensive mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was initially created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the contending Num array into Numeric, with a wide range of variations. NumPy is open-source software and has numerous contributors.

**3.4 MATPLOT:**

Mat plot is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits such as wxPython, Tkinter, Qt, or GTK. There is also an established "Pylab" interface grounded on a state machine (like OpenGL), designed to relate/appear like that of MATLAB, though its use is dejected. SciPy makes use of Matplotlib. It was initially written by John D. Hunter.

**3.5 SKLEARN:**

Scikit-learn (Sklearn) is the most helpful and vigorous library for machine learning in python. It provides a selection of economical and productive tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistent interface in python. It was firstly called scikits.learn and was originally developed by David Cournapeau as a Google summer of code project in 2007.

**3.6 SEABORN:**

Seaborn is a wonderful visualization library for statistical graphics plotting in python. It provides beautiful default styles and color palettes to make these plots more attractive. It is erected on top matplotlib library and is nearly integrated with data structures from pandas. It will be mainly used to visualize arbitrary distributions.

**3.7 PANDAS AND NLTK:**

Pandas is a python library utilized for working mainly with datasets. It has methods for inspecting, cleaning, exploring and controlling data. The title Pandas has a mention of both “Panel Data” and “Python Data Analysis” and was created in 2008. On the other hand, Natural Language Toolkit (NLTK) is a python programming environment for generating applications for statistical NLP.

After having a brief description of the packages used in this project, Now, we will move on to the algorithm used in the project which are as follows: -

**3.8 SUPERVISED LEARNING:**

As we already know that machine learning is of three types and here, in this project we will substantially deal with supervised learning. In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of training set), and then it predicts the output. Supervised learning model is used here because it helps us to solve numerous real-word problems such as fraud detection, spam filtering, etc. which are mostly similar to the text classification problem. Moreover, Supervised learning can be further divided into two types of problems: Regression and Classification.

**3.9 DECISION TREE ALGORITHM:**

Decision Tree is a basically a supervised learning method that can be utilized for both Categorization and Regression problems, but mostly it is popular for tackling Classification problems. It is a tree-structured classifier, where internal nodes represent the characteristics of a dataset, branches represent the decision rules and each leaf node represents its output. In a Decision tree, there are two mainly 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. It is a graphical representation for getting all the possible solutions to a problem or decision grounded on given conditions. **It is known as a decision tree because, similar to a tree, it originates with the root node, which extends on further branches and constructs a tree-like structure.**

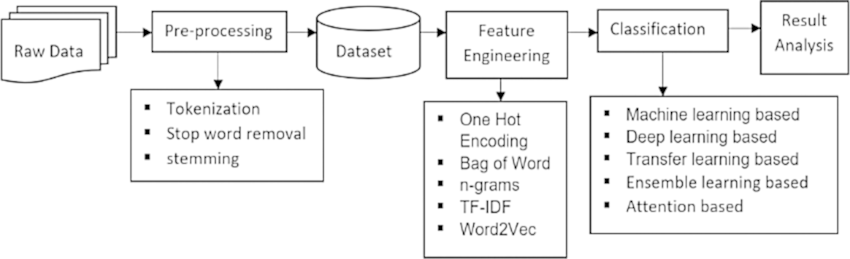
Now, The Question Arises:

→**WHY THE SUPERVISED LEARNING ALGORITHM WHICH IS MENTIONED ABOVE IS PREFFERED THAN THE OTHER ALGORITHMS?**

It is because the kind of project which we are working on is based on classification problem which means that there are classes/categories which we are required to predict. In this type of problem, the output variable is categorical. Since the above-mentioned algorithm which is

Decision Tree comes under the category of classification-based algorithms and is well suited for our project design; therefore, we are applying them.

**3.10 SYSTEM ARCHITECTURE:**



**Fig. 3.1 System Architecture of Project [5]**

**3.10.1 Importing Libraries and Dataset:**

In the first step, we import the essential libraries like NumPy, Seaborn, etc. and also the dataset titled as ‘twitter\_data.csv’ (which was downloaded from internet) to the jupyter notebook so that we can use the dataset and resources available in these libraries required for the further process.

**3.10.2 Creating a Stemmer and a set of Stopwords:**

In the coming step, we will use the snowball stemmer to perform the stemming process as it is a better version of the Porter Stemmer. Stemming is important for natural language processing and is used for reducing a word to its base word. Also, we create a set of stopwords like and, or, not, etc. which we will remove later on at the time of text cleaning.

**3.10.3 Mapping Numerical Class Labels to Text Labels:**

In this step, we map the numerical class labels to text labels so here, ‘0’ represents ‘Hate Speech Detected’, ‘1’ represents ‘Offensive Language Detected’, ‘2’ represents ‘No Hate or Offensive Speech’ and now all the text present in the dataset will be categorized to these labels except the NaN values.

**3.10.4 Creating a Function to Clean Text:**

In this step, we will manually define a cleaning function in which we will remove any kind of unnecessary text and all the stopwords, after that we will assign these new values to tweet labels.

**3.10.5 Converting Raw data/text into Matrix and Splitting Dataset for further process:**

Text to vectors is an approach that involves converting text data into numerical formats suitable for use by machine learning algorithms. This is significant because Machine learning algorithms can only work with numerical inputs and by using the CountVectorizer we can achieve that. After that, we will split the dataset into training set (67%) and test set (33%) to prevent the problem of overfitting.

**3.10.6 Creating the Model and Checking its Accuracy:**

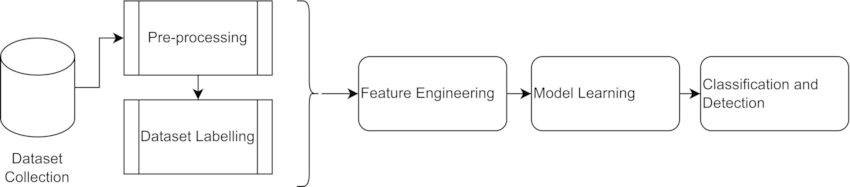
In this step, we will create our model using the supervised machine learning algorithm (Decision Tree) and then fit/train the model using the train data. Then we will create labels for the test data and ultimately check the accuracy of the trained model by taking the new data or using the test data along with it we will print its classification report.

**3.10.7 Creating the Confusion matrix and Heat Map:**

In this step, we will create a heat map of the confusion matrix using the seaborn library and then plot it as x-axis representing as ‘True Label’ and y-axis representing as ‘Predicted Label’. Also, we will use this heat map to verify the accuracy of the trained model.

**3.10.8 Categorization of the Text:**

This is the last and the final step in which we will define a function for our model to check whether it is capable of detecting the speech entered to it. We will check by manually inputing the text as a string using the ‘predict\_text’ function.



**Fig. 3.2 Block Diagram of System Architecture [6]**

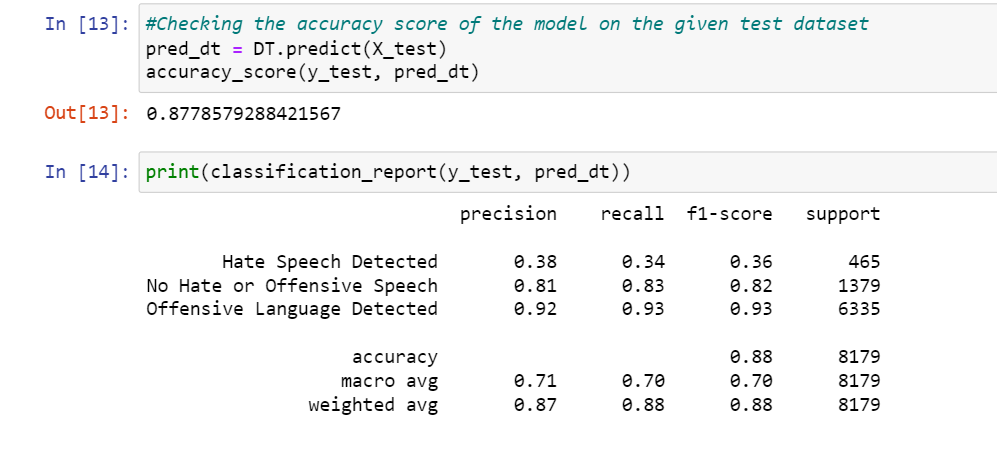
**Chapter 4**

**Result and Discussion**

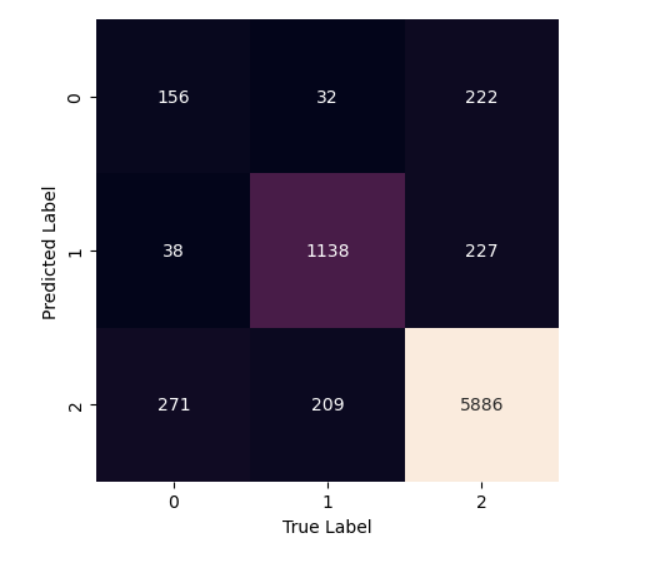
My project is able to detect the hate speech and categorize it with almost 88% accuracy (87.78% to be precise) on the basis of dataset which we have taken from the internet. Below are the outputs of the sample texts which have been predicted and categorized accurately.



Moreover, this hate speech detection code is being run on Jupyter Notebook and the accuracy of the model along with the classification report is being given as follows: -



Also, the Confusion Matrix for the project is depicted below where ‘0’ represents ‘Hate Speech Detected’, ‘1’ represents ‘Offensive Language Detected’ and ‘2’ represents ‘No Hate or Offensive Speech’.



**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion:**

In this report, an innovative model for hate speech detection using classification-based algorithm has been presented. The model which we have made is producing accurate results as can be seen from the output screenshots in the result and discussion section above. Considering the accuracy of the model which was almost 88% (based on the prediction of model), we can say that machine learning can be used as an effective tool for detecting hate speech and helping to combat it through early detection. Hate Speech Detection using Machine learning algorithms is a promising approach that can be used in business organizations, MNC’s, etc. Machine learning algorithms can analyze huge datasets and identify patterns that are commonly found in hate speeches, offensive languages, speeches igniting violence/violent behaviour, etc. and hence, can be used to accurately predict the nature of speech for the data/text tagged with it.

**5.2 Future Work:**

Presently, the model which we are using for hate speech detection is based on machine learning algorithms and is working by feeding the datasets, on which it is being trained and is capable of identifying and categorizing the kind of speech, but they are not entirely reliable, as there is always a possibility of misidentification of one kind of speech with other and vice-versa. So, the use of machine learning for the detection of hate speech is still in its early phase. Deep learning algorithms and other machine learning algorithms like naïve bayes and support vector machine can be considered as an alternative approach for this project. In future, using concise in-built functions and advanced AI systems, we could use a hate speech detection system which will be able to classify any kind of speech and no matter how much complex the data is, it would be able to predict the results immediately and most importantly it would not be limited to a particular dataset. Also, currently the developed model is working at 88% efficiency so, in future, the new model may be able to achieve almost 90-99% accuracy.

**References**

[1] Hate Speech Detection Functionalities:

http://imsypp.ijs.si/

[2] Multi-modal Hate Speech Detection using Machine Learning

December 2021

DOI:10.1109/BigData52589.2021.9671955

Conference: 2021 IEEE International Conference on Big Data (Big Data)

Authors: Fariha Tahosin University of Conneccticut Ponkoj Shill University of Nevada, Reno

Md Golam Rabiul Alam BRAC University

[2] Hate speech detection: A comprehensive review of recent works February 2024 Expert Systems DOI:10.1111/exsy.13562

Authors: Ankita Gandhi Param Ahir Kinjal Adhvaryu Pooja Shah

[3] Hate Speech Detection using Machine Learning

Publisher: IEEE

P. Preethy Jemima; Bishop Raj Majumder; Bibek Kumar Ghosh; Farazul Hoda

Date of Conference: 22-24 June 2022 Date Added to IEEE Xplore: 29 July 2022

DOI: 10.1109/ICCES54183.2022.9835776 Publisher: IEEE

[3] Z. Zhang, L. Luo

Hate speech detection: A solved problem? the challenging case of long tail on twitter

Semantic Web, 10 (2019), pp. 925-945.

[4] Hate Speech Detection Using Decision Tree Algorithm

J. Lavanya M. Ramesh J. Sravan Kumar G. Rajaramesh \* Subhani Shaik

Published: 2023-06-19

DOI: 10.9734/jamcs/2023/v38i81791

Page: 66-75

Issue: 2023 - Volume 38 [Issue 8]

Bao Y. and Ishii N., “Combining Multiple kNN

Classifiers for Text Categorization by

Reducts”, LNCS 2534, 2002, pp. 340-347

[5] System Architecture of Project:

https://www.researchgate.net/publication/378479636\_Hate\_speech\_detection\_A\_comprehensive\_review\_of\_recent\_works

[6] Block Diagram of System Architecture:

https://www.researchgate.net/publication/378479636\_Hate\_speech\_detection\_A\_comprehensive\_review\_of\_recent\_works

**THANKYOU!**