Multi-modal Hate Speech Detection Using Machine Learning

Ponkoj Chandra Shill
16201011
Fariha Tahosin Boishakhi
16201010

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ponkoj Chandra Shill
16201011

Fariha

Fariha Tahosin Boishakhi
16201010

Approval

The thesis titled “Multimodal Hate Speech Detection Using Machine Learning” submitted by

1. Ponkoj Chandra Shill (16201011)
2. Fariha Tahosin Boishakhi (16201010)

Of fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Fall, 2020.

Examining Committee:

Supervisor:
(Member)

Md Golam Rabiul Alam, PhD

. Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac Universitv

Mahbubul Alam Majumdar, PhD

Professor and Dean, School of Data and Sciences
Department of Computer Science and Engineering
Brac University

Abstract

Hate speech is a common problem that people face in any content based applications. With continuous growth of internet users and media contents, it is very hard to track down hateful speech in audio and video. Converting video or audio into text does not detect hate speech accurately as humans sometimes use not hateful words as hate speech in a sarcastic way and also uses different voice tone or shows different action in the video than text. In the research, a combined approach to detect hate speech from contents using video, audio and speech by extracting feature images, feature values extracted from audio, text and used Machine learning, Deep learning and Natural language processing to detect hate speech.

Keywords: Audio hate Speech , Video hate Speech, Hate Speech detection, Machine Learning, Multi-modal Hate Speech detection.

Acknowledgement

We are grateful to our thesis supervisor, Dr.Golam Rabiul Alam, for his consistent guidance and to make us envision that this research is possible. It is through him that we realized the significance of this topic, and to detect hate Speech more effectively.

For our parents, without their support throughout, it would not be have been possible to reach this far. With their kind support and prayer, we are now on the verge of completing the final phase of our thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion,sex, or sexual orientation, states by Cambridge Dictionary. Now-a-days hate speech are not only in social media comments and posts or text messages but also in the voice messages and video contents too. It might be some disrespectful pictures, memes, videos, voice messages.It is seen that people are becoming more creative with hate speech. They use hate speech in a way that does not contain offensive words but still manages to be hate speech. These type of contents not only spread hatred and misinformation but also results in crimes like cyber bullying, riot, fraud, loss of respect and even murder. In 2020, a movement "Black Lives Matter' took place in order to bring justice to the death of Gorge Floyd who died from police brutality. The evidence of the protest was a video taken by strangers passing across the street. So, it can be seen how videos effects our lives not only for entertaining purpose but also to connect people.

1.2 Motivation

The United State Department of Justice , FBI released hate crime statistics which showed in 2019, 15,588 law enforcement authorities provided information on the crimes, suspects, perpetrators and crime regions. There were 7,314 crime cases including 8,559 offense speech among these organizations that submitted the reports.Among these reports 57.6 percent are Race/Ethnicity/ Ancestry/bias, 20.1 percent are religion based, 16.7 percent are sexual orientation based. Another report on online hate crime describes that hate crimes usually start from online and they affect us offline. Victims of the report stated that they were scared of physical safety and were abused in the real world and offline. A report shows the top five ways to spread hate speech online and among them social networks, video platforms are the top two. So, not taking into account any video or audio feature to detect hate speech creates a hole in the detection system. So,In order to prevent such crime, the first step is to detect such contents so it does not spread such hate messages.[25][30][18][32]

1.3 Problem Statement

After researching on how hate speeches are delivered in different content is related to voice tone and expression. It was seen that the same speech with a different tone is used for expressing different meanings. For example, “I hate you” is not only used to express hatred but also used among friends to express friendliness while cracking a joke. In a friendly conversation, people use slang that is not considered as hate speech rather a friendly conversation. Text based detection can not detect such cases as it does not contain the audio feature or image feature to identify the exact expression that the content is conveying.

1.4 Objectives

The objective of the research is to detect hate speech from any kind of content and specially from videos. All the recent work that has been done on hate speech detection is mostly on text data. This research showed a new approach to detect hate speech using audio and video features and also using text features. With the feature extracted from the data hate speech is detected more precisely.

1.5 Work Plan

The work plan set up to do the research had some steps. First we analyzed how hate speech was conveyed generally and what contents we should focus on. After analyzing we found the three most prominent ways of conveying hate speech. We collected the dataset and converted them into different types of data. We found the factors and features from each type of the dataset. We then collected features and applied various machine learning algorithms and deep learning methods to detect the content is hateful or not using each of their results differently and then combining them together giving the final result.

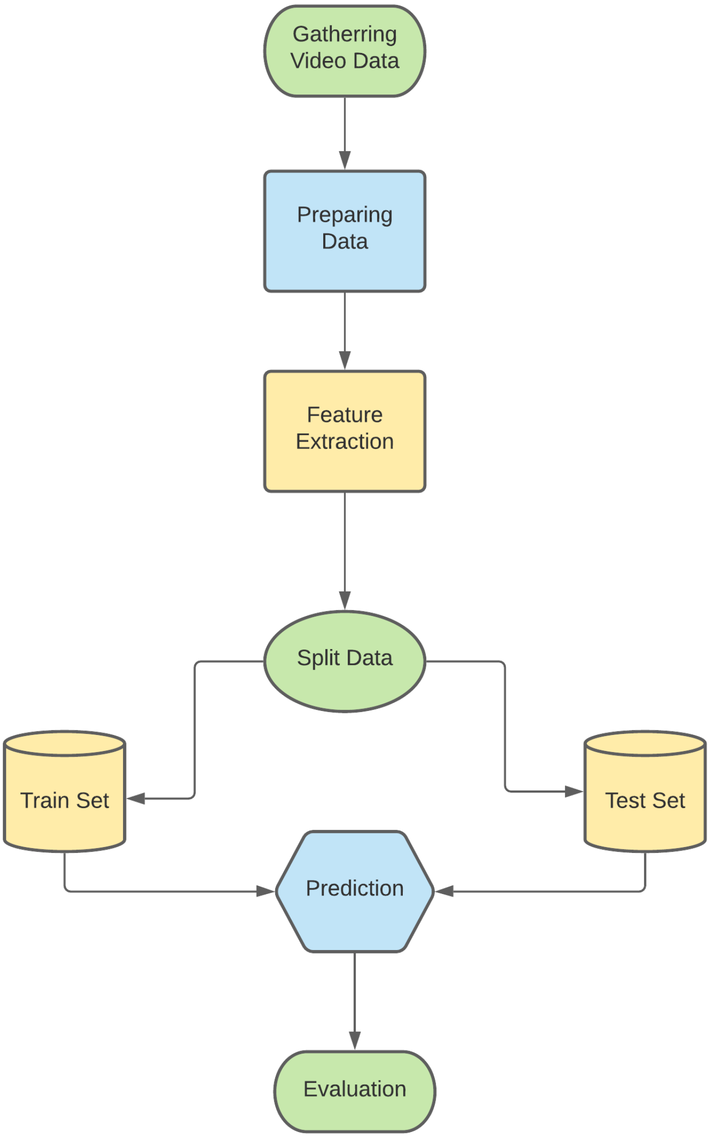


Figure 1.1: Workplan for Multimodal Hate Speech Detection

1.6 Thesis Overview

The rest part of this thesis report has been arranged as follows:

- Chapter 2, analyzes a thorough study of most related works regarding our thesis work.
- Chapter 3 describes the dataset we collected from different sources to detect Multi-modal Hate Speech Detection.
- Chapter 4 discusses the various algorithms and classifiers applied in our research, and the accuracies obtained and compares the algorithms and analyzes the results obtained and comments on the performance..
- Chapter 5 concludes our thesis and proposes our future plan regarding our research work.

Chapter 2

Literature Review

2.1 Related Works

People often use hate words or contents through social or entertaining content and regular life as well. There is enough research based on tweeter or text to detect hate content where a very few research work is done to detect hate speech from video or audio contents. We have found and investigated research work based on text, audio and video for our research purpose.

We investigated an application of deep neural network architectures in order to do hate speech detection from tweets. Abusive speech targeting- cyber bullying, Gender, religion and LGBT are classified and filtered as racist, sexiest or neither. Various DNN methods are used (CNN, LSTM, Fast Text) to identify which one is the best. They have experimented with different type classifiers like logistic regression , SVM , random forest classifier and deep neural networks. They used CNN [Convolutional Neural Network] for sentiment classification, LSTM [Long Short Term Memory] for capturing long range dependencies, Fast-Text for averaging word vectors and back-propagation to fine tune the representation. They used 16k annotated tweets with respect to GLoVe pre trained word embedding. They have used 'Adam' algorithm in CNN and LSTM on the other hand 'RMS-props' in fast-text for optimization purpose. For baseline approach ,the TF-IDF implementation is performing better than those of the character n-gram algorithm. Similarly, random embeddings works perfectly with the help of GBDT. They did many tests with many methods but LSTM with random embedding working with deep neural network models and gradient boosted decision trees gave the best accuracy values. Prec-0.930, Recall-0.930, F1- 0.930. On the other hand, Combinations of CNN, LSTM, FastText embeddings as features for GBDTs was not that much better to be appreciated.[10].
Audio segmentation is basically used to preprocess the data. In this paper they have introduced different well known segmentation methods. They focused on the Log-linear model to determine segmentation from audio streams. There is also comparison of different speech recognition techniques. For segmentation, the non-speech is detected and removed (music and other unused sounds) and overlapping speeches are separated to reduce the error rate. They used three methods- 1.decoder guided segmentation which is used to silent regions recognized, 2. model based approach which is used to classify regions divide audio stream, 3. Metric base approach –distance between audio streams used. They used ASR model along with matric

based approach along with a different segmentation approach. The front line consists of MFCC and VTLN(Vocal Tract Length Normalization). To recognize unsegmented data, data wrapping factors processed independent acoustic model with speaker. Using the segment clusters, CMLLR feature transformations and MLLR mean transformations are evaluated. They did mention the use of sum of negative logarithm for implementation. They analyzed some feature- segment length, words, boundary length, boundary confidence, wrapping factor variance, BIC score, signal type, sentence end. In training, they used GIS(Generalization Iterative Scaling) for optimization. To detect speech they used 3-state HMMs for the signal type audio, non-speech, pure music or tune and silence. They used Gaussian mixtures as emission models. The mixture models were trained with 5 classes labelled for 9.5hrs audio material. For individual signals, they estimated a bigram language model. For the dataset, they used American English broadcast news data. They presented a MAP decoder framework for audio segmentation.Among 15s segments, NIST, ASR-based and MAP, MAP is the best for audio signals they found. So, in our paper we must use audio segmentation in the preprocessing step for these reasons[1].
Detection of online based Hate Speech Using Context Aware Models, two methods were used- logistic regression with context characteristics and neural network with context of learning. It is different as other methods do not contain contextual hate speech recognition. The results found from testing cases are identical. Accuracies are -Char(baseline)-0.738 , Best Neural Network Model-0.766 and best logistic regression model-0.750. So, the average we found is 0.779. Logical regression model shows 3% better result and neural model shows 4% better result. Both showed 7% better results to recognize hate speech. So, we can use detecting implicit and creative hate content and detecting long comments with regional focus of hateful speech or words [35].

Another research, they have presented isolated Hindi isolated words recognition where speech recognition system understands a voice by microphone and converts it into text. Their main motivation is to recognize speech from isolated words for Hindi language. They have taken a large amount of voice data from native people manually for the purpose of making a dataset for their research work. For feature extraction they have used MFCC technique and vector quantization with GMM [Gaussian Mixture Model] for detection of hindi isolated words. They prepared hindi word speech dataset of different males and females speakers for their practice analysis. In the training mode they processed utterances and found the feature vector of corresponding utterances using MFCC feature extraction technique. To obtain MFCC vectors they followed some preliminary pre-processing steps such framing, windowing, Fourier Transformation, Mel Frequency Rapping, discrete cosine transform and calculation of Delta [] and it's coefficients. This is the way they prepared their training model. For external utterances, isolated words are given as input via microphone. In the testing process external utterances taken as input and matched against the training model. In this step, KNN [K-Nearest Neighbours] is used for pattern classification. The overall result is quite satisfactory for isolated hindi words. A ninety point zero nine percent [90.09%] precision, night-eight point three eight [98.38%] percent recall and ninety-four point three one percent [94.31%] accuracy prove that this system can be used for speech recognition. As most of the automatic speech recognition systems are mainly based on English language, speech detection systems in Hindi language can be an important means of education for disable and illiterate people and the communication system of the whole country.[9]

For detecting hate speech from websites containing hateful words, epithets, phrases, stereotypical thought, anti- racial content using SVM [Support Vector Machine] and it had an accuracy of 94% in classifying anti-sematic speech, 68% in precision, 60% in recall and F1 at a measure of .6375. The dataset was collected from Yahoo!! and American Jewish Congress.Average length of the data was 31 although some were more. Datas containing one word or more than 64 words, incomplete sentences, contained two or more unicode characters was ignored. An amount of 9000 paragraphs was matched by using a general regular expression of words related to Judiatsm and Israel and another 1000 paragraphs were labeled by annotators. The paragraphs then were labeled into 7 different categories- Anti Semitic,anti-feminism, Anti-Asian,Anti-black, anti-immigrants, anti-Muslim or other hate. Another set called gold was created for removing error. For classification approach they used template based strategy and each template was centred around single words and this process produced 4379 features. Using an SVM classifier model the data was fed and by eliminating features they got 3537 features. Two additional sets containing 272 features were found using the unigram set.The set contained 13 features and the most significant is “television”. A baseline was set (N-Np)/N which yielded a baseline accuracy of 0.910. Six classifies was created to classify.The research successfully determined the hateful content of different websites.[6]
Detecting hate speech from text on social media, some challenges have to be faced like nuance and language subtleties, different definitions of what constitutes hate speech, data availability limitations, interpret ability problems. As a review paper, they have reviewed some other papers regarding hate speech from text on social media. Therefore, they have proposed mSVM [Multi-View Support Vector Machines] model for this purpose, having some other models. The only motivation of their work is automatic detection of hate speech that can reduce the spread of hateful content and speech. They have indicated some popular dataset such as HateBase Twitter, WaseemA, WaseemB, Stormfront, TRAC, HatEval, Kaggle. After all they have reviewed the keyword based approach for hate speech detection. For keyword based approach, this approach is fast and straightforward but has some false alarming problems such as trash, swine etc. Additionally, they have reviewed the Source Metadata approach which needs additional information like demographics platform, place, timestamp or social engagement that can give further understanding of hate recognition. But this approach has the big issue of unavailability of this kind of data from the authority for external researchers. Another approach is Machine Learning Classifiers, Neuman for feature extraction, Naive Bayes, SVM, Logistic regression for text categorization, Davidson for feature based classification in the form of TF-IDF, Neural ensemble, FastText, BERT, C-GRU. Finally, the proposed model mSVM where they have used case-folding, tokenization, removing punctuations as preprocessing. For feature extraction, they have extracted TF-IDF from unigram to 5-gram. They also have done character N-gram from unigram to 5-gram. They used Stamfron, TRAC, HatEval HateBase Twitter dataset. Finally, the evaluation result showed for StormFront dataset BERT having 0.8201 percent accuracy and mSVM having 0.8033 percent accuracy, for TRAC dataset BERT having 0.5809 percent accuracy and mSVM having 0.6121 percent accuracy. As having some challenges,

further research on this topic, covering both technological and practical issues is required. [26]
The goal of this research is to find out how NLP [Natural Language Process] contributes detection of hate speech. Here they followed five major steps of Natural Language Process- Classification : text classifying and sentiment analysis, Matching: matching two strings, Translation:speech recognition and machine translation , Structured Prediction: mapping string to certain structure, Sequential Decision Process: taking actions in changing environment. According to their research, CNN [Convolutional Neural Network] was far better than LSTM [Long-Short Term memory] and FastText where LSTM is better than FastText. They used a dataset of hate speech identification colected from CloudFlower. The dataset contains 24786 English tweets where tweets are categorized into three classes. There are 1430 hate speech related tweets, 19190 tweets contain offensive words, 4163 tweets contain neither of them. These amounts indicate 5% hate speech and 77% offensive which makes the dataset quite imbalancing. All of the experiments have been done in python and they used Machine Learning and neural Network libraries. Furthermore, the model could not identify many tweets accurately. Besides, it misclassified 80% of the hate classes due to the unbalanced training data. Though it identifies many non-hate speech as hate speech classes, it obtains quite good performance [91% accuracy]. As most of the tweets were from offensive class, the model leads to be biased towards classifying tweets as offensive. However, if dataset is richer in quality and size CNN can provide good performance.[16]
The goal of this paper is to classify online hate speech in general and also particularly racist and sexist speech on deep learning methods using pre-trained word embeddings and max/mean from simple, fully connected transformations of embeddings. Three most widely used datasets were used. These datasets were named as Sexist/Racist(SR) collected by Waseem and Hovy(2016), HATE collected by Davidson et al(2017) , HAR collected by Golbeck et al(2017). After that These datasets were labelled in following terms- sexist racist, either; hate speech or not ; harassment or not. The training set consisted of N examples X, Y=1, where the input is X and Y is the label or numerical class of the hate speech. In addition to that SWEM-concat was used to handle frequent and unknown words and capture nonlinear word combinations. They used 300 dimensional embeddings, 1-layer Multi Layer perceptron Wt with a Relu activation to form an updated embedding space z1:T. Two pooling methods were used, one is max pooling to capture salient word feature from input noted as m and another is to averaging embeddings z1:T to capture over all meaning of the sentences, denoted as a. As for Preprocessing they experimented with different approaches while tokenizing the data using Spacy. Using 300 dimensional Glove Common Crawl embeddings and fine tuning the data. The data was padded each input into 50 words. They used RMSprop with the learning rate 0.001 and batch size 512.The results are produced from 10- fold cross validation to allow comparison of previous data.The baseline for HAR dataset was taken out using a logistic Regression model.The result showed more efficiency than many complex methods which in total is F1 12 points. They suggested studying the occurrence of such linguistic phenomena in existing dataset and construct new dataset that have better representation of subtle forms of hate speech.[20]
Annotating hate speech reliably is one of the main jobs to detect hate speech. As the definition of hate speech varies from person to person, so some people might

not find some speech offensive while others do. So to annotate hate speech reliability Our results indicate showing users a definition caused them to partially align their own opinion with the definition but did not improve reliability,which was very low overall. We conclude that the presence of hate speech should perhaps not be considered a binary yes-or-no decision, and raters need more detailed instructions for the annotation. Three different approaches were taken, but the reliability did not differ from different approaches even after the given defination.In the end it was suggested that Researchers who are building a hate speech detection system might want to collect multiple labels for each tweet and average the results. [13]

We investigated a review paper and we got to know about the techniques that were used in various stages of automatic speech recognition.Among these, basic building blocks is the primary step of speech and language process.In this step natural speech is converted into modified speech and symbol sequence. Then feature and parameter are extracted via feature extractor. After that features turn into labeled segment via segmenter and labeler.The next stage is classification of speech recognition system.Words can be classified into several categories such as isolated,connected and continuous.Speech recognition system can be classified as discrete or continuous that asr speaker dependent or independent.In discrete speech system speaker needs to stop between the words, but a continuous speech system face difficulties to find the start and the end point of word. That's why discrete speech is quite easier than the other one.Furthermore size of vocabulary can be small, medium or large which can affect the complexity of the system.They also mentioned three different approaches for speech recognition, they are acoustic phonetic approach, pattern recognition and Artificial intelligence. After all the paper provided the comprehensive survey of research in speech recognition system.[5]

We found a research work where the author has used multiple machine learning algorithms to detect spam from SMS. For this he used the dataset from UCI Machine Learning repository. In preprocessing, he simply just tokenized the data based on alphabetic characters where any special character or space, dot, comma are simply removed and the abbreviation-misspelling is ignored. No stop words, stemming or lemmatization algorithms are used in this research. In the training period he used multiple classification models such as Naive Bayes, SVM(Support Vector Machine), KNN, Ensemble method called Random Forests and Adaboost. Finally, he states that for that certain dataset the best classifier is SVM with 97.64% accuracy and followed by Naive Bayes with overall 97.50% accuracy.[36]

In a research, we found emotion recognition based on audio and text using multimodal neural networks. They used CNN and LSTM combined to learn emotional features from audio. Furthermore, they used Bi-LSTM(Bidirectional Long Short-term Memory) Networks to extract the text features. Then they applied a DNN to learn and classify the features.[22]

Chapter 3

Methodology

In this chapter, we have provided details of the data collection process, data labelling and refining stage, dataset validation and the test data.

3.1 Data Collection

Collecting the data is the first and most important task for the research. As the research is trying to detect hate speech targeting video content, we collect the data from different movies, web series and youtube videos containing hateful conversation, chaos and fight.The dataset is taken from different popular sites such as youtube.
The video dataset is taken as length of one, at most two full sentences where the hate speech occured.The length of the videos ranges from 1 to 5 seconds.The frame rate of the videos are 30.
The dataset is labeled in two categories - Hate and Non Hate dataset.The video dataset containing hate speech with aggressive or hateful expressions are labeled as Hate and non hateful expressions are labeled as Non Hate.
The video dataset is converted into three types of Dataset - Image, Audio and Text
.

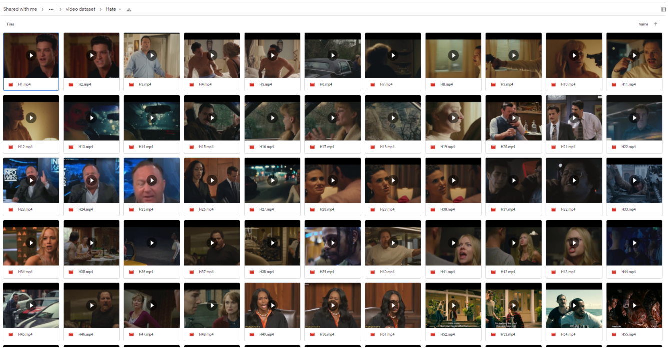


Figure 3.1: Video Data Collection

3.1.1 Image Dataset

The video dataset is converted into Images taking each of the frames into account. Using OpenCV each of the frames is captured. The images are labeled as Hate and Non Hate according to their contents.

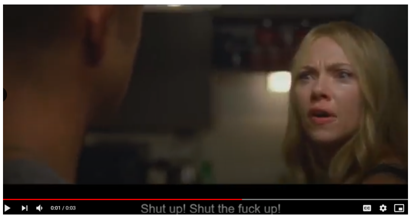


Figure 3.2: Image Data

3.1.2 Audio Dataset

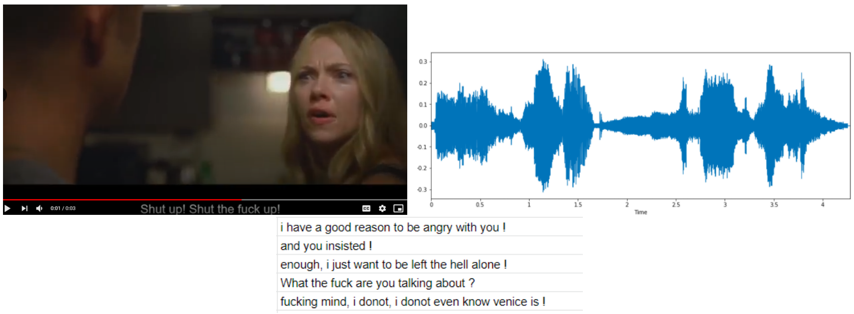
Within the auditory spectrum available to humans, audio is sound. An audio frequency is an electrical alternating current that can be used to create acoustic sound ranging from 20 to 20,000 hertz. Sound is a collection of ordinary analog signals which are converted by the audio card to digital signals using a microchip known as analog-to-digital converter.When sound is played, digital signals are sent to the microphones, where they are converted back to analog signals that create varied sounds.So audio is unstructured data and, by default, audio signals are non-stationary, meaning their properties differ over time (usually rapidly). The more extreme samples captured dominate the resulting value. Instead, it would be more efficient to break the record into short segments and calculate one (average) strength value per segment. This is also the central principle behind short-term processing. The video dataset is converted into audio using and we reduce the background noise and remove the unnecessary noise.

3.1.3 Text Dataset

The Audio dataset was converted into Text using Google Speech API. Each of the dataset was uploaded to the cloud and converted into a text and labeled as Hate and Non Hate accordingly. After converting the dataset we collected, we see most of the datas has one and at most two sentences so we had to do tokenization. In tokenization, it takes the uppercase character to lowercase characters and splits the sentences. After that it takes the Sentences and removes the punctuations with the help of a regular expression . The tokens are then lemmatized or stemmed. Stemming is closely associated with lemmatization. The difference is that without awareness of the context, a stemmer works on a single word and can therefore not distinguish between words that, depending on part of the expression, have different meanings. Stemmers, however, are usually simpler to implement and run quicker, and for some applications, the decreased precision does not matter. For this one we used both of the processes differently. After that we removed the stop words, Stop words are such words that do not affect the meaning of the sentences much.

3.2 Feature Extraction

we extracted features from all the dataset seperately



i have a good reason to be angry with you !
and you insisted !
enough, i just want to be left the hell alone !
What the fuck are you talking about ?
fucking mind, i donot, i donot even know venice is !

Figure 3.3: Image, Audio and Video Data Preprocessing

3.2.1 Image Features

For the image dataset, As the images are categorized into two categories Hate and Non-Hate. We converted the dataset into pickle files where each of the data is converted into an array.Images of the videos are taken and resized 50 *50. As the images are nothing but series of array. We are taking all of the images from a video and sorted and label accordingly to extract the image features.

3.2.2 Audio Features

In the audio dataset, we need to set up a sampling rate as the sampling rate or sampling frequency determines the number of samples taken from a continuous signal per second (or per unit) to make a discrete or digital signal. Frequencies are calculated in hertz (Hz) or cycles per second for time-domain signals such as the waveforms for sound (and other audio-visual content types). The Sample rate for the following task is 22050 as we found in Google Speech recognition and they suggested a frequency more than 16 KHz.
we went through different sources to look for feature extraction.[14] For audio feature extraction two types of features are extracted from the audio signal. One is Time Domain Features and the other is Frequency Domain Features.[8]

Time Domain Features

For time domain features The Short term processing is mentioned earlier shows we need to divide the audio in different frames to process further and for that we took hop length of 256 and frame length of 512 to divide the signal for sampling. From each of the frames we took the following feature and Calculated the mean of the output.[32]

Energy

Energy can be described as the signal's strength. It is possible to calculate the signal power by its energy. Energy is determined by the sum of signal squares, normalized by frame values the signal's duration.

$$E(i) = \sum_{n=1}^{W_L} |x_i(n)|^2 \tag{3.1}$$

Zero-Crossing Rate

This feature is selected for the following task as it gives the overall idea of the strength of the signal which gives better results in speech related research. Zero-Crossing Rate: An audio frame's Zero-Crossing Rate (ZCR) is the rate of signal sign-changes during the frame. In other terms, it is the number of times that the Signals shift the value, divided by the frame length, from positive to negative and vice versa. Signals shift the value, divided by the frame length, from positive to negative and vice versa. The following equation determines the ZCR according to -

$$Z(i) = 1/2W_L \sum_{n=1}^{W_L} |Sgn[x_i(n)] - sgn[x_i(n-1)]| \tag{3.2}$$

where sgn() is the sign function,

$$Sgn[x_i(n)] = 1, x_i(n) \geq 0, -1, x_i(n < 0 \tag{3.3}$$

Frequency Domain Features

Spectral Centroid

The spectral centroid is the spectrum's 'gravity' nucleus. It means where a sound is located in the' center of mass' and is measured as the weighted mean of the frequencies present in the sound. The spectral centroid will be around a middle if the frequencies of an audio are the same throughout, and if at the end of the sound there are high frequencies, then the centroid will be at its end.

Spectral Rolloff

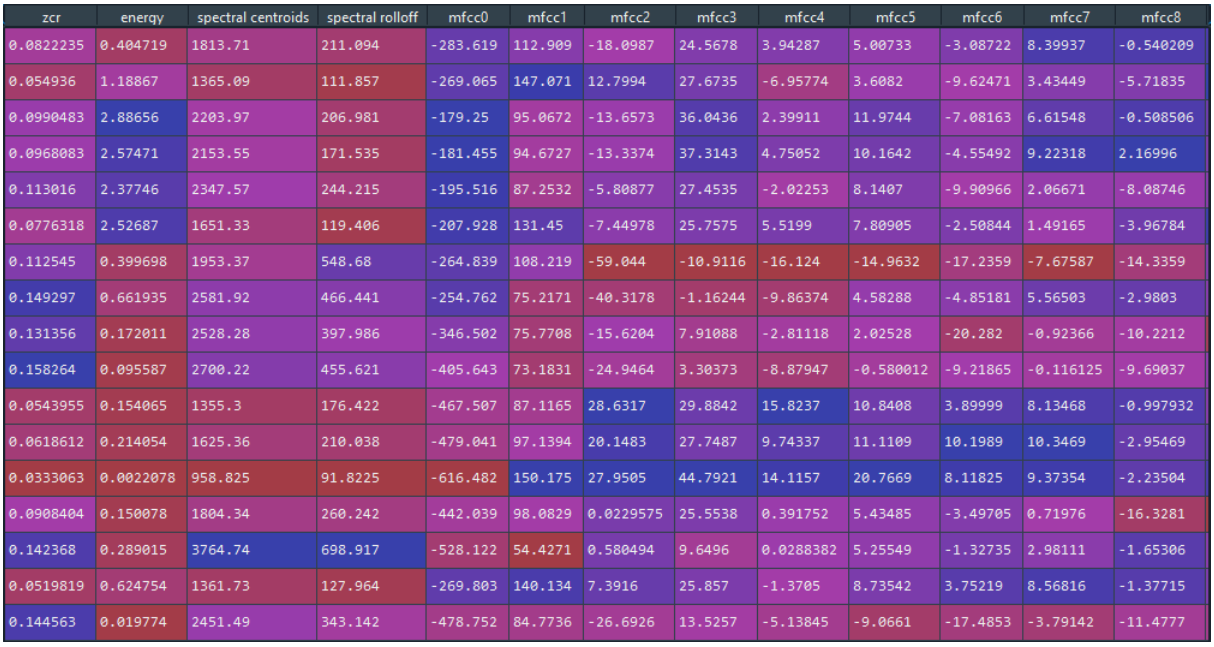
Spectral rolloff measures the frequency below which a specified (usually 90%) percentage of the total spectral energy.

Mel Frequency Cepstral Coefficient (MFCC)

A signal's mel-frequency cepstral coefficients (MFCCs) are a limited number of characteristics that concisely define a spectral envelope's overall structure. As MFCC acts on the Fourier transformation of a signal, using triangular overlapping windows, it maps the above spectrum powers to the mel size.At any of the mel frequencies, it takes the log value of the powers to take the discrete cosine transform of the mel log powers list as if it were a signal. The MFCCs are the resulting frequency amplitudes. In automatic speech and speaker recognition, MFCCs are a feature commonly used.Different extraction features were used prior to the implementation of MFCCs, but they had different sets of noise reduction and accuracy issues, but mfcc was better than both of them. We took 20 of the MFCC elements,

Chroma Vector

The Chroma vector would be a 12-element spectral energy representation. The chroma vector is calculated by combining the DFT coefficients of a short-term window into Twelve bin. Every bin reflects one of the 12 equal measured pitch groups of the data input. Every bin represents the mean of the respective DFT coefficients' log magnitudes.



zcr	energy	spectral centroids	spectral rolloff	mfcc0	mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	mfcc6	mfcc7	mfcc8
0.0822235	0.404719	1813.71	211.094	-283.619	112.909	-18.0987	24.5678	3.94287	5.00733	-3.08722	8.39937	-0.540209
0.054936	1.18867	1365.09	111.857	-269.065	147.071	12.7994	27.6735	-6.95774	3.6082	-9.62471	3.43449	-5.71835
0.0990483	2.88656	2203.97	206.981	-179.25	95.0672	-13.6573	36.0436	2.39911	11.9744	-7.08163	6.61548	-0.508506
0.0968083	2.57471	2153.55	171.535	-181.455	94.6727	-13.3374	37.3143	4.75052	10.1642	-4.55492	9.22318	2.16996
0.113016	2.37746	2347.57	244.215	-195.516	87.2532	-5.80877	27.4535	-2.02253	8.1407	-9.90966	2.06671	-8.08746
0.0776318	2.52687	1651.33	119.406	-207.928	131.45	-7.44978	25.7575	5.5199	7.80905	-2.50844	1.49165	-3.96784
0.112545	0.399698	1953.37	548.68	-264.839	108.219	-59.044	-10.9116	-16.124	-14.9632	-17.2359	-7.67587	-14.3359
0.149297	0.661935	2581.92	466.441	-254.762	75.2171	-40.3178	-1.16244	-9.86374	4.58288	-4.85181	5.56503	-2.9803
0.131356	0.172011	2528.28	397.986	-346.502	75.7708	-15.6204	7.91088	-2.81118	2.02528	-20.282	-0.92366	-10.2212
0.158264	0.095587	2700.22	455.621	-405.643	73.1831	-24.9464	3.30373	-8.87947	-0.580012	-9.21865	-0.116125	-9.69037
0.0543955	0.154065	1355.3	176.422	-467.507	87.1165	28.6317	29.8842	15.8237	10.8408	3.89999	8.13468	-0.997932
0.0618612	0.214054	1625.36	210.038	-479.041	97.1394	20.1483	27.7487	9.74337	11.1109	10.1989	10.3469	-2.95469
0.0333063	0.0022078	958.825	91.8225	-616.482	150.175	27.9505	44.7921	14.1157	20.7669	8.11825	9.37354	-2.23504
0.0908404	0.150078	1804.34	260.242	-442.039	98.0829	0.0229575	25.5538	0.391752	5.43485	-3.49705	0.71976	-16.3281
0.142368	0.289015	3764.74	698.917	-528.122	54.4271	0.580494	9.6496	0.0288382	5.25549	-1.32735	2.98111	-1.65306
0.0519819	0.624754	1361.73	127.964	-269.803	140.134	7.3916	25.857	-1.3705	8.73542	3.75219	8.56816	-1.37715
0.144563	0.019774	2451.49	343.142	-478.752	84.7736	-26.6926	13.5257	-5.13845	-9.0661	-17.4853	-3.79142	-11.4777

Figure 3.4: Audio Feature Extraction

3.2.3 Text Features

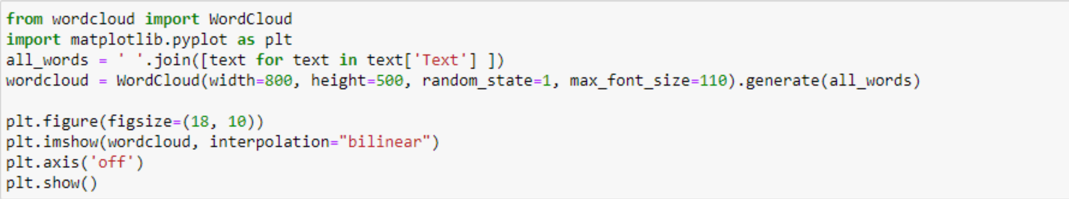
Before feature extraction we use some preprocessing steps to make the text dataset precise and effective for our proposed model. First of all, tokenization steps are taken to tokenize the speech into single sentences.In this step we use sentence tokenizer. After tokenizing the whole speech or sentences we eliminate some regular expressions or special characters (like . , _ @ space etc). We just take uppercase and lowercase alphabet (a-z and A-Z). We make all the words into lowercase this time. After these preprocessing the data we then used the following ways to convert the data to vectors-

Counter Vectorizer/Bag of Words(BOW)

he model of bag-of-words (BOW) is a representation that transforms arbitrary text into vectors of fixed length by counting the number each word appears. Here, based on the label of the dataset and each word of a single data or sentence are given values.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical test that assesses in a series of documents how important a term is to a text. This is achieved by multiplying two metrics: how many times a term in a document appears, and the word's inverse document frequency over a range of documents.[21] [31] [23] The formula that is used to compute the tf-idf is : TF-IDF = TF * IDF



```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
all_words = ' '.join([text for text in text['Text'] ])
wordcloud = WordCloud(width=800, height=500, random_state=1, max_font_size=110).generate(all_words)

plt.figure(figsize=(18, 10))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

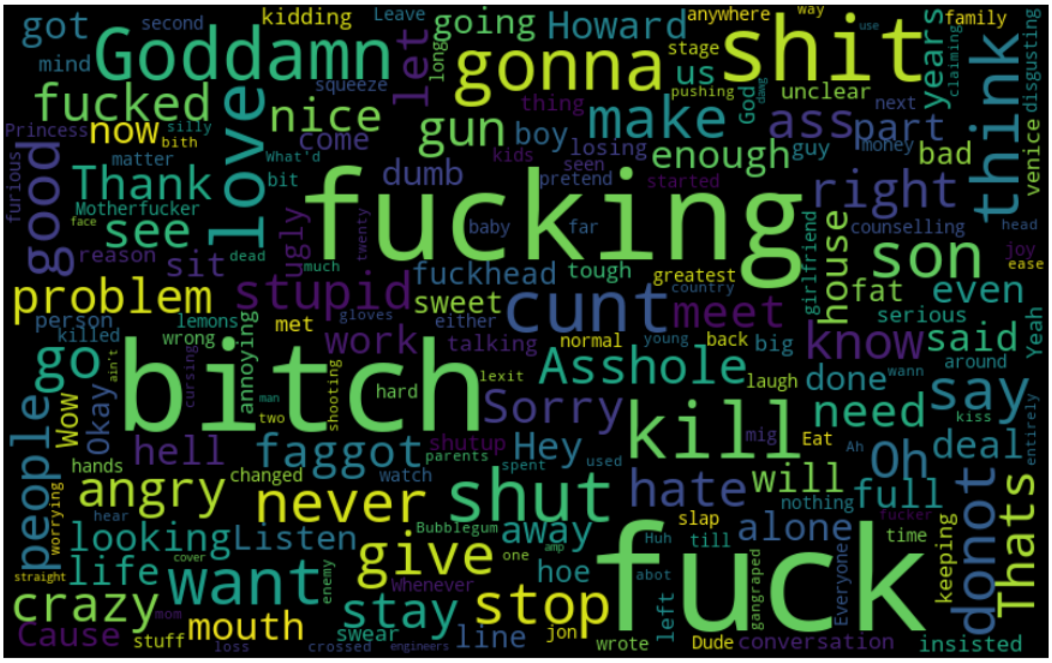
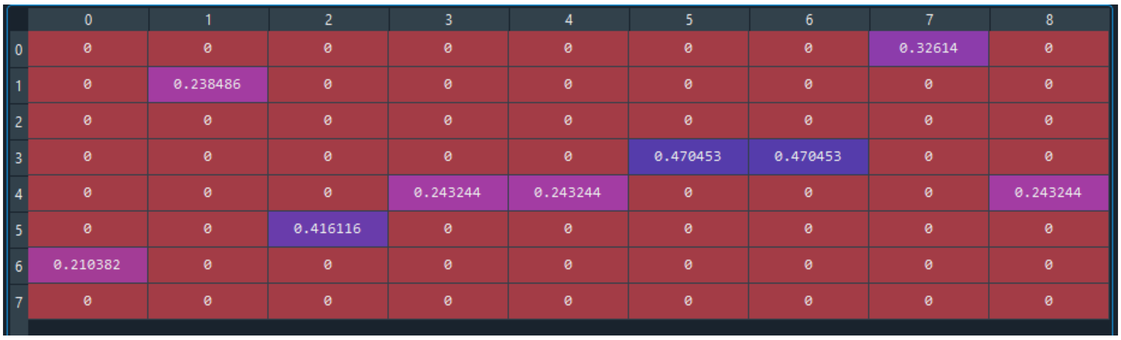


Figure 3.5: Most Frequent words In Text Dataset

Where, EquationTF = (Number of time the word occurs in the text) / (Total number of words in text)
IDF = (Total number of documents / Number of documents with word t in it)



	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0.32614	0
1	0	0.238486	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0.470453	0.470453	0	0
4	0	0	0	0.243244	0.243244	0	0	0	0.243244
5	0	0	0.416116	0	0	0	0	0	0
6	0.210382	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0

Figure 3.6: Feature Extraction with TF-IDF

3.3 Proposed Multimodal Hate Speech Detection method

This section descibes the The methodology for this research is centralized on detecting hate speech from all the major communication platform possible, After Collecting and Extracting the data, we will be taking one single algorithm and fit the values of the features. Each of the features, image, audio and text will be passed separately to find out the accuracy of the model. The prediction we will get from the algorithm will be taken and then go through a majority voting ensemble to predict the final output.[3]

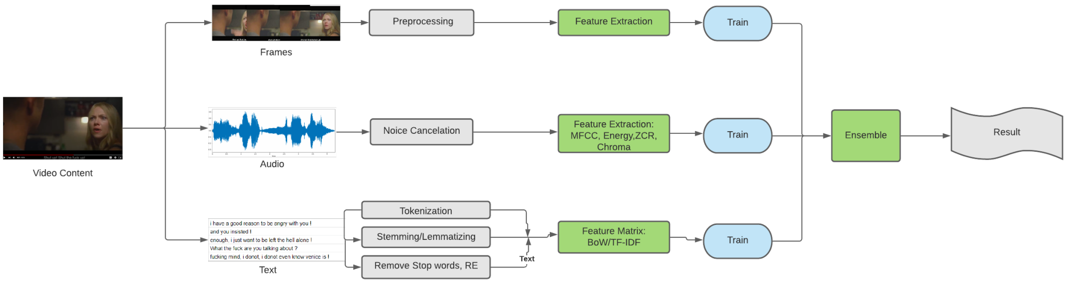

Frames
Preprocessing
Feature Extraction
Train
Video Content
Audio
Noice Cancelation
Feature Extraction: MFCC, Energy,ZCR, Chroma
Train
Ensemble
Result
i have a good reason to be angry with you !
and you insisted !
enough, i just want to be left the hell alone !
What the fuck are you talking about ?
fucking mind, i donot, i donot even know venice is !
Text
Tokenization
Stemming/Lemmatizing
Remove Stop words, RE
Text
Feature Matrix: BoW/TF-IDF
Train

Figure 3.7: proposed Multimodal Hate Speech Detection method proposal

3.4 Model Specification

In this section, we will look at the different machine learning algorithms that we have used to determine the results.

3.4.1 Support Vector Machine

The Support Vector Machine, a supervised machine learning algorithm that can be used for classification or regression problems. However, it is often used in classification problems.

Supervised learning that analyzes data used for analysis of classification and regression. SVM varies in the way that it selects the decision boundary that maximizes the distance from the closest data points of all classes from the other classification algorithms. An SVM doesn't just discover a decision boundary, it chooses the most optimal decision boundary. Such points are called vectors of support. Now, the distance between the line and the support vectors is computed. The margin is called this gap. Our purpose is the maximization of the margin. The ideal hyperplane is the hyperplane for which the margin is maximum.

In the SVM algorithm, it maps each data object as a point in n-dimensional space (where n is the number of characteristics you have) with the value of a certain coordinate being the value of each characteristic. We carry out classification by finding the hyper-plane that differentiates the two classes very well.Support Vectors are essentially independent observation coordinates. The SVM classifier is a boundary that divides the two groups most effectively (hyper-plane/line).

We are looking to optimize the margin between the data points and the hyperplane in the SVM algorithm. Hinge loss is the loss function which helps optimize the margin.

If the expected value and the real value are of the same signal, the cost is 0. We then determine the loss value if they are not. A regularization parameter is also applied to the cost function. The goal of the regularization parameter is to offset the maximization and lack of margin. The cost functions appear as below after adding the regularization parameter.

With reference to the weights, we take partial derivatives in order to locate the gradients. We are able to change our weights using the gradients.

It is easy to have a linear hyper-plane between these two groups in the SVM classifier. But, another burning issue that emerges is, in order to have a hyper-plane, we need to add this function manually. No, the SVM algorithm has a so-called kernel trick strategy. The SVM kernel is a functionality that takes up and translates low dimensional input space into higher dimensional space.

3.4.2 Random Forest

For supervised learning, Random Forest is an algorithm. The "The " it creates is an ensemble of decision-making trees, usually trained in the " it constructs is an ensemble of decision trees, usually trained in the method of " method. The fundamental concept of the bagging approach is that a combination of learning types increases the combined result.

Random Forest is a supervised learning method that is used for both classification and regression. But it is, however, mainly used for classification problems.A forest consists of trees, and more trees make the forest stronger. Similarly, on data samples, the random forest algorithm generates decision trees and then gets the result from each of them and chooses the best solution by voting. It is an ensemble strategy that is greater than the individual decision tree, since it avoids over-fitting by integrating the outcome.

In one sentence, random forest constructs and merges several decision trees to achieve a more detailed and stable forecast. Random forest is a perfect algorithm to train, to see how it operates, early in the model creation process. Its simplicity makes the creation of a ”bad” random forest a challenging proposition. Also, the algorithm is a perfect alternative for someone who wants to rapidly create a concept. On top of that, it gives a reasonably clear idea of the weight it attributes to your characteristics.

Another wonderful quality of the random forest algorithm is that the relative significance of each function on the forecast is very simple to calculate. Sklearn offers a fantastic tool for this, which tests the value of a feature by looking at how much impurity is minimized among all trees in the forest by the tree nodes that use that feature. After practicing, it measures this score automatically for each role and scales the scores such that the sum of all significance equals one.[12][29]

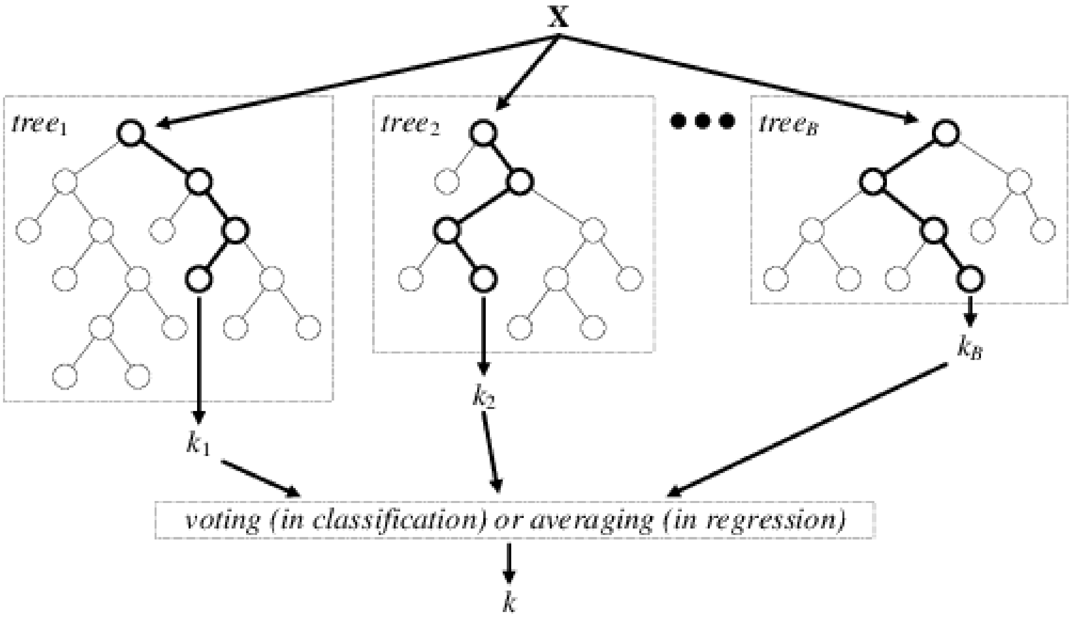


Figure 3.8: architecture of Random forest

3.4.3 K Nearest Neighbour

KNN is a lazy learning algorithm that is non-parametric. Non-parametric implies that no expectation exists for the distribution of underlying results. The model structure, in other words, was calculated from the dataset. In reality, this would be very useful where most of the real world datasets do not meet theoretical mathematical assumptions. The Lazy algorithm implies that model generation does not require any training data points. Both details on the instruction used in the evaluation process. This makes preparation quicker and the process of testing slower and more expensive. The expensive process of research means time and memory. The core decision factor is the number of neighbors. K is normally an odd number, because the number of classes is 2.[24] [33]

KNN can be used both for predictive problems of classification and regression. It is, however, most generally used in industry classification issues. We normally discuss three critical things in order to determine any technique:1. Facility for interpreting output, 2. Time to calculate,3. Power of prediction.

3.4.4 Naive Bayes Classifier

Naive Bayes classifiers are a set of Bayes Theorem-based classification algorithms. It is not a single algorithm, but an algorithm family where a common concept is shared by all of them.
It is a grading methodology based on the Bayes theorem, which implies that predictors are independent. Simply placed, a Naive Bayes classifier believes that there is no relation to the existence of a certain function in a class.[19]

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})} \qquad ||I, \tag{3.4}$$

The Gaussian model assumes that a normal distribution is followed by features. This means that the model assumes that these values are sampled from the Gaussian distribution if predictors take continuous values instead of discrete ones.

3.4.5 Logistic Regression

A statistical model used to evaluate if an independent variable has an effect on a binary dependent variable is logistic regression. This implies that given an input, there are only two possible results.

$$1/(1 + e^{-}value)$$

(3.5)

When the dependent variable is binary, logistic regression is the best regression analysis to perform. Logistic regression can be used to characterize details Logistic regression is named for the equation used at the core of the method, the logistic function. Statisticians developed the logistic function, also called the sigmoid function, to explain biological population growth characteristics, growing rapidly and maxing out at the environmental carrying power. It is just an S-shaped curve that can take any real value number and map it to a 0 to 1 value, and not precisely at those limits.[27]

3.4.6 Decision Tree

The algorithm for theThe Decision Tree is part of the group of algorithms for supervised learning. The decision tree algorithm might be used for solving classification and regression, as comparison to many other supervised learning algorithms.
In Decision Trees, we start from the root of the tree to predict a class label for a text. We compare the values of a root attribute with the attributes of the record. We follow the branch corresponding to that value on the basis of comparison, and jump to the next node. A decision tree is a map of the future consequences of a set of connected decisions. It helps a person or organisation, depending on their costs, possibilities and advantages, to weigh future actions together. They may either be used to lead casual conversations or to map an algorithm that mathematically predicts the right choice.

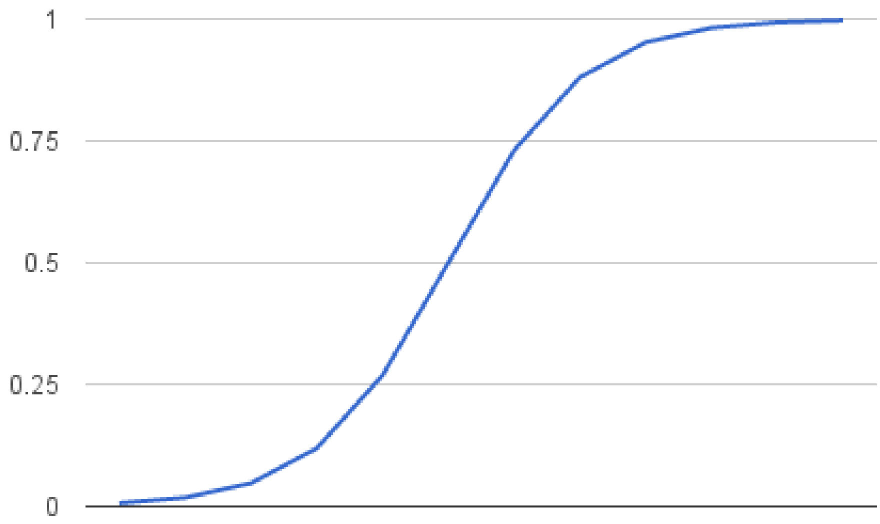


Figure 3.9: Logistic Function

Usually, a decision tree begins with a single node that divides into potential consequences. Any result leads to additional nodes that connect with other alternatives. It has a treelike form. A decision tree can also be used to construct automatic statistical models for machine learning, data processing and statistics applications. This approach takes into account assumptions on an item known as decision tree learning to estimate the meaning of the item.
Nodes reflect data rather than judgments in these decision-making bodies. The classification tree is sometimes called this type of tree. Each branch has a collection of characteristics or rules for classification that is related to a basic class mark located at the bottom of the branch.These laws sometimes called the rules of judgment, for each decision or data value that constitutes a clause, can be represented by an if-then clause where, for example, "if conditions 1, 2 and 3 are fulfilled, the outcome x will result with certainty." Every additional data piece helps to predict more specifically which values belong to the specific subject. This knowledge will then be used as an input to a wider paradigm of decision making. The projected variable often is a real number, like a price. Regression trees are called decision trees with constant, limitless potential effects.[21]

3.4.7 Adaboost Classifier

Ada-boost or Adaptive Boosting is one of Yoav Freund and Robert Schapire's descriptions of ensemble boosting in 1996. It incorporates many classificators to make the classificators more precise. AdaBoost is a grouping system that is iterative. By merging many badly performing categories, AdaBoost classifier creates a powerful category to give you high precision strong categorisations. The underlying principle behind Adaboost is that the weights of classifiers and the data sample should be conditioned in each iteration in such a way that unexpected observations are expected accurately. If a machine learning algorithm accepts weights on the training set it can be used as a simple classificator. Two conditions for Adaboost: the classificator should be interactively conditioned on different examples of weighted instruction and it attempts to render the best match for these examples by reducing the exercise error in every iteration.

AdaBoost classifier is a meta-estimator, starting at the moment where the classifier is placed into the initial data and later fits copies that are additional of that classifier into the same dataset, which changes the weights of incorrct classification instances to concentrate more on complicated cases afterwards.

At first, Adaboost arbitrarily picks a training subset. It is learned iteratively by choosing the training set based on a specific forecast for last training for the Adaboost machine learning model. It gives greater weight to inaccurate categorized observations so that these observations are more likely to be classified in the next rotation. In each iteration, it also assigns the weight to the trained classifier according to the exactness of the classifier. The more specific the weight classifier is. This method is carried out until the entire course data meets the defined maximum number of estimators without errors or has been hit. To identify, make a "vote" across all of the study algorithms that have been created.[2] [7] [28]

Chapter 4

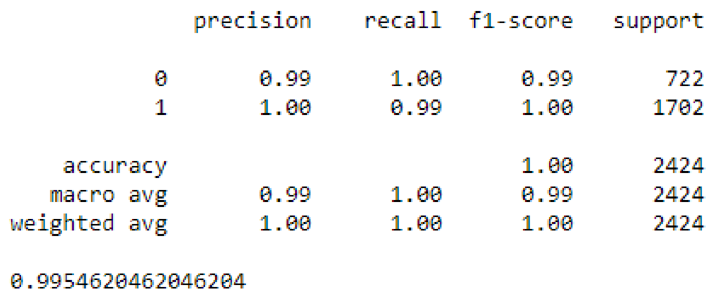
Performance Analysis

We applied Image, Audio and Text features in all the algorithms separately to classify Hate and Non Hate and then used Majority Voting Based Ensemble Classification at the end to find out the Final result.

4.1 Support Vector Machine

4.1.1 Image Dataset

As the algorithm is a supervised learning algorithm and we are using it to classify two classes the SVM gives out a very delicate result. As Image classification is a method of image processing which, according to the different characteristics of images, distinguishes between different target categories the SVM takes the input and creates the best hyperline. We took the kernel polynomial as It is widely used with SVM and other kernelized models, which represent the similarity of vectors in the original variables' feature space over polynomials, allowing non-linear models to be learned.[17] [11]



```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       722
           1       1.00      0.99      1.00      1702

    accuracy                           1.00      2424
   macro avg       0.99      1.00      0.99      2424
weighted avg       1.00      1.00      1.00      2424

0.9954620462046204
```

Figure 4.1: The Classification Report of Matrix using Support Vector Machine for image

4.1.2 Audio Dataset

The audio dataset feature values are passed into SVM with the polynomial kernel which helps to categorize the non linear models to run. As the values of the dataset contains values of different features from some audio signals which carries the discrete values of a signal. After applying the audio dataset it creates a hyperplane to

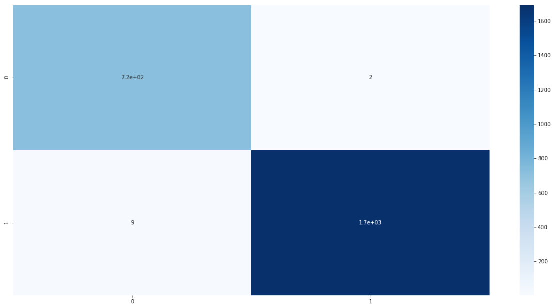
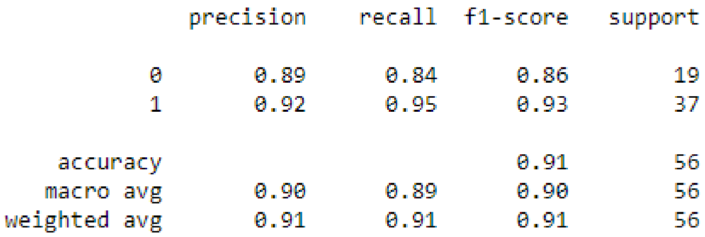


Figure 4.2: The Confusion Matrix of Support Vector Machine for image

determine the classification of a single record of the dataset. Each of the records are dedicated to determine one value.



```
              precision    recall  f1-score   support

           0       0.89      0.84      0.86        19
           1       0.92      0.95      0.93        37

    accuracy                           0.91        56
   macro avg       0.90      0.89      0.90        56
weighted avg       0.91      0.91      0.91        56
```

Figure 4.3: The Classification Report of Matrix using Support Vector Machine for Audio

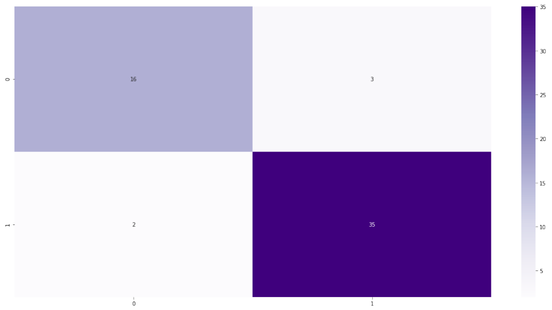
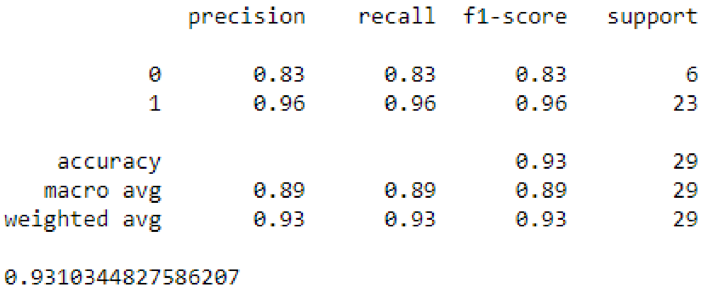


Figure 4.4: The confusion matrix of Support Vector Machine for Audio

4.1.3 Text Dataset

After Preprocessing the dataset shows most frequent words and after applying SVM, it detects the hate and non hate data from the dataset. A hyperplane is created with a polynomial kernel. Keeping the feature vectors into account shapes the dataset it minimizes the error of the fitted values.



```
              precision    recall  f1-score   support

           0       0.83      0.83      0.83         6
           1       0.96      0.96      0.96        23

    accuracy                           0.93        29
   macro avg       0.89      0.89      0.89        29
weighted avg       0.93      0.93      0.93        29

0.9310344827586207
```

Figure 4.5: The Classification Report of Matrix using Support Vector Machine for Text

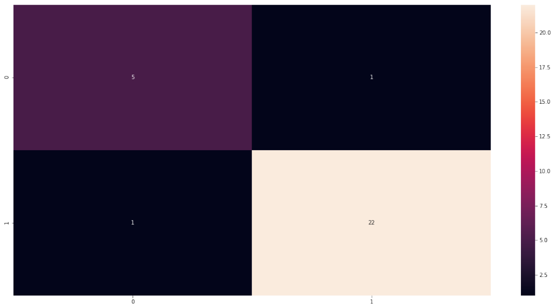
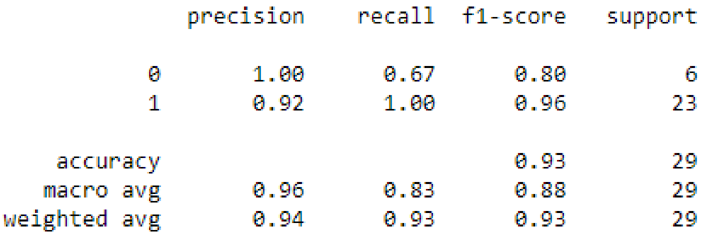


Figure 4.6: The Confusion Matrix of Support Vector Machine for Text

4.1.4 Ensemble Result

The ensemble tecnic is used to incease the accuracy based on the output of SVM result previously gotten. Among the result of the previous results we are going to take the best output of two classifiers gives positive result then that output will be selected.



```
              precision    recall  f1-score   support

           0       1.00      0.67      0.80         6
           1       0.92      1.00      0.96        23

    accuracy                           0.93        29
   macro avg       0.96      0.83      0.88        29
weighted avg       0.94      0.93      0.93        29
```

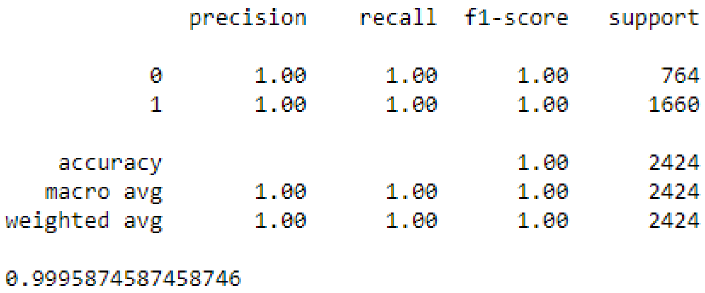
Figure 4.7: The Classification Report of Matrix using Support Vector Machine for Ensemble

4.2 Random Forest

4.2.1 Image Dataset

As the algorithm is a supervised learning algorithm and we are using it to classify two classes the Random forest gives out a very delicate result. As Image classification is a method of image processing which, according to the different characteristics of images, distinguishes between different target categories the Random forest takes

the input and creating the base case and then proceeding with a simple yes or no clause. As random forest takes the inputs of the images which are arrays and fitted into each of the base cases to determine the record is hated or not. But we can see the results given here are very high due to the dataset we choose. We will work to minimize the values of the error rate and avoid the overfitting conditions.
After applying random forest in the image dataset we see there is 99.5% accuracy.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       764
           1       1.00      1.00      1.00      1660

    accuracy                           1.00      2424
   macro avg       1.00      1.00      1.00      2424
weighted avg       1.00      1.00      1.00      2424

0.9995874587458746
```

Figure 4.8: The Classification Report of Matrix using Random Forest for image

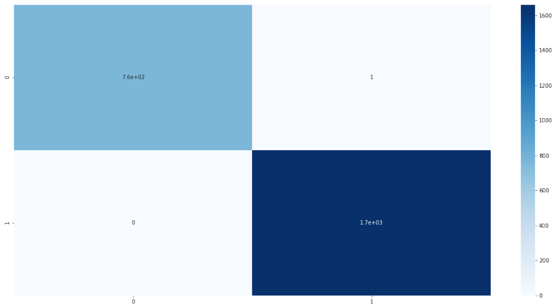
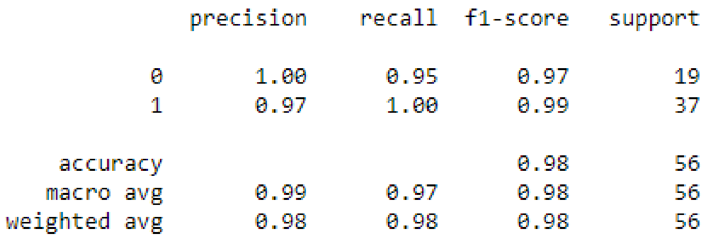


Figure 4.9: The Confusion Matrix of Random Forest for image

4.2.2 Audio Dataset

The feature values of the audio dataset are transferred to Random Forest, which helps to categorize hate and non-hate by selecting the values and generating a random record base case. Because the data set values contain values of different characteristics from some audio signals that bear the separate signal values, each record is dedicated to the determination of one value and converted to the labels.



```
              precision    recall  f1-score   support

           0       1.00      0.95      0.97        19
           1       0.97      1.00      0.99        37

    accuracy                           0.98        56
   macro avg       0.99      0.97      0.98        56
weighted avg       0.98      0.98      0.98        56
```

Figure 4.10: The Classification Report of Matrix using Random Forest for Audio

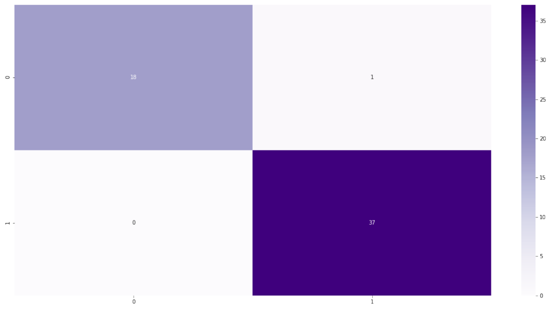
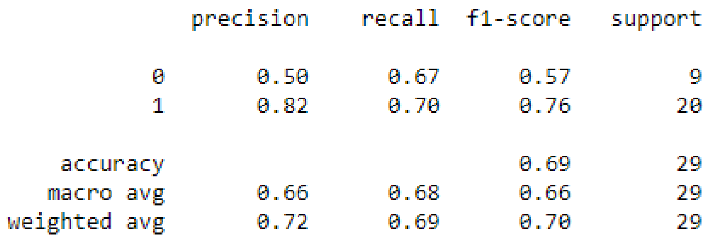


Figure 4.11: The confusion matrix of Random Forest for Audio

4.2.3 Text Dataset

After Preprocessing the dataset shows most frequent words and after applying Random forest, it detects the hate and non hate data from the dataset. Random forests are a very unreliable classifier, especially if only a few features are very necessary for the outcome to be decided. In this matrix, all TF-IDF features are important, so the random forest performs fine when predicting the result.



```
              precision    recall  f1-score   support

           0       0.50      0.67      0.57         9
           1       0.82      0.70      0.76        20

    accuracy                           0.69        29
   macro avg       0.66      0.68      0.66        29
weighted avg       0.72      0.69      0.70        29
```

Figure 4.12: The Classification Report of Matrix using Random Forest for Text

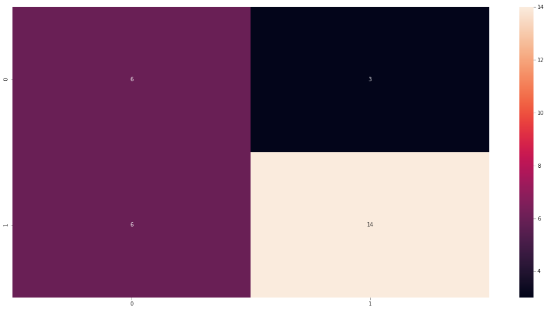
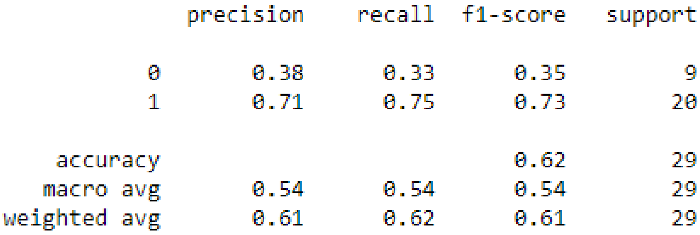


Figure 4.13: The Confusion Matrix of Random Forest for Text

4.2.4 Ensemble Result

Taking the output of three initial outputs to confirm the final output of hate or non hate.



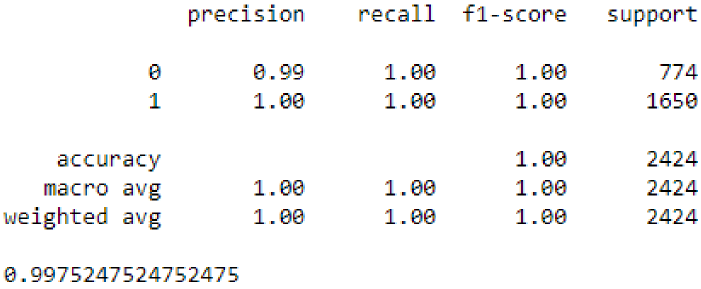
```
              precision    recall  f1-score   support

           0       0.38      0.33      0.35         9
           1       0.71      0.75      0.73        20

    accuracy                           0.62        29
   macro avg       0.54      0.54      0.54        29
weighted avg       0.61      0.62      0.61        29
```

Figure 4.14: The Classification Report of Matrix using Random Forest for Ensemble

4.3 K-Nearest Neighbor

By defining the most prevalent class of the k-closest cases, the k-NN algorithm classifies unknown data points. Each data point casts a vote in the nearest k and the group with the most votes wins. The Image datasets, audio dataset and text datasets are classified into two classes using the default classification for 2.

4.3.1 Image Dataset



```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       774
           1       1.00      1.00      1.00      1650

    accuracy                           1.00      2424
   macro avg       1.00      1.00      1.00      2424
weighted avg       1.00      1.00      1.00      2424

0.9975247524752475
```

Figure 4.15: The Classification Report of Matrix using K-Nearest Neighbor for image

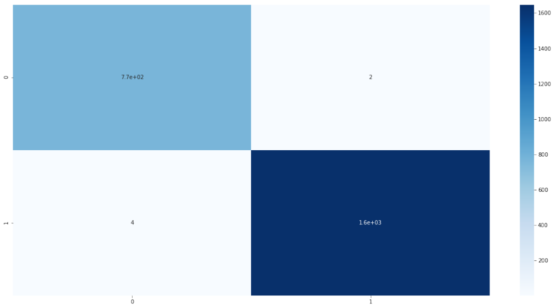
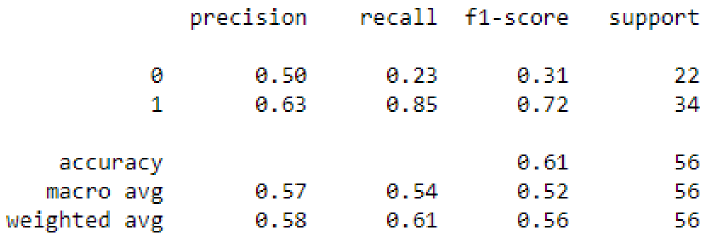


Figure 4.16: The Confusion Matrix of K-Nearest Neighbor for image

4.3.2 Audio Dataset

We got the accuracy of 69% after running the Audio dataset.



```
              precision    recall  f1-score   support

           0       0.50      0.23      0.31        22
           1       0.63      0.85      0.72        34

    accuracy                           0.61        56
   macro avg       0.57      0.54      0.52        56
weighted avg       0.58      0.61      0.56        56
```

Figure 4.17: The Classification Report of Matrix using K-Nearest Neighbor for Audio

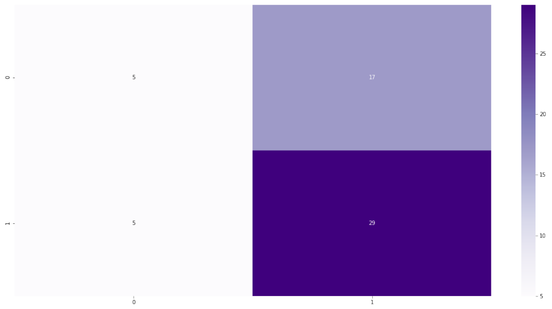
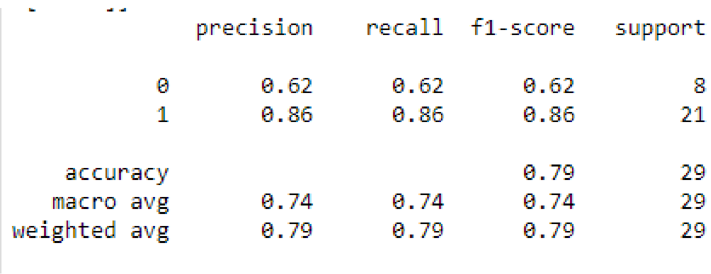


Figure 4.18: The confusion matrix of K-Nearest Neighbor for Audio

4.3.3 Text Dataset

For text dataset we got the accuracy of 79



```
              precision    recall  f1-score   support

           0       0.62      0.62      0.62         8
           1       0.86      0.86      0.86        21

    accuracy                           0.79        29
   macro avg       0.74      0.74      0.74        29
weighted avg       0.79      0.79      0.79        29
```

Figure 4.19: The Classification Report of Matrix using K-Nearest Neighbor for Text

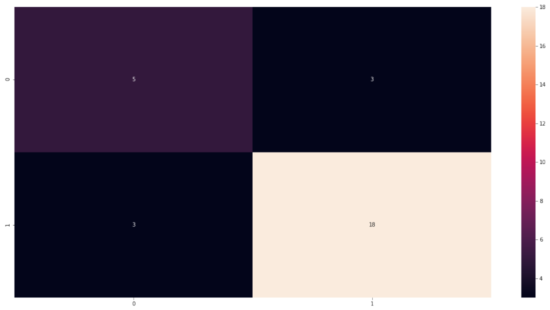
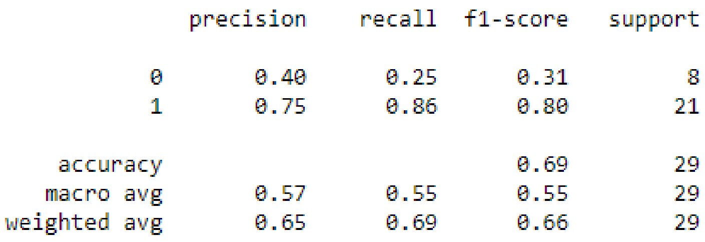


Figure 4.20: The Confusion Matrix of K-Nearest Neighbor for Text

4.3.4 Ensemble Result

After Ensambling we got 69% accuracy



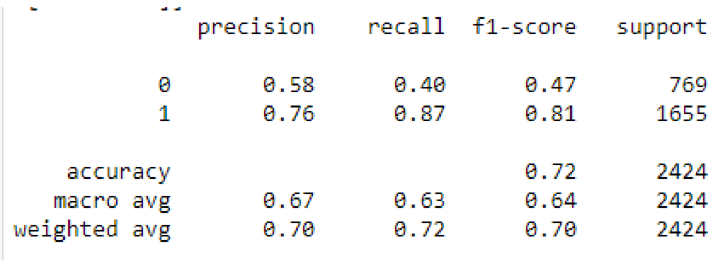
```
              precision    recall  f1-score   support

           0       0.40      0.25      0.31         8
           1       0.75      0.86      0.80        21

    accuracy                           0.69        29
   macro avg       0.57      0.55      0.55        29
weighted avg       0.65      0.69      0.66        29
```

Figure 4.21: The Classification Report of Matrix using K-Nearest Neighbor for Ensemble

4.4 Naive Bayes

4.4.1 Image Dataset

Naive Bayes is, however, a classifier in probabilism. The class or mark that the algorithm learns to predict is the function of creating the probability distribution of all the classes it is shown, and then choosing which one to assign to each case. Probabilistic classifiers analyze the distribution of conditional probabilities, taking into account each of the image. Using the array of images the classifies predicts the probability of each classes to identify the outcome. [15]



```
              precision    recall  f1-score   support

           0       0.58      0.40      0.47       769
           1       0.76      0.87      0.81      1655

    accuracy                           0.72      2424
   macro avg       0.67      0.63      0.64      2424
weighted avg       0.70      0.72      0.70      2424
```

Figure 4.22: The Classification Report of Matrix using Naive Bayes for image

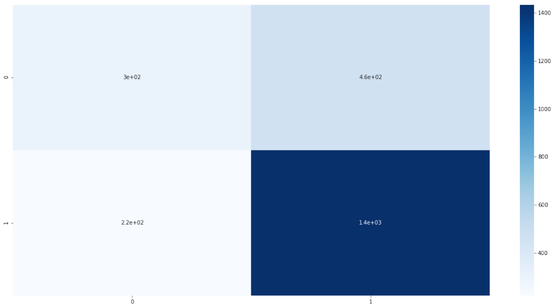
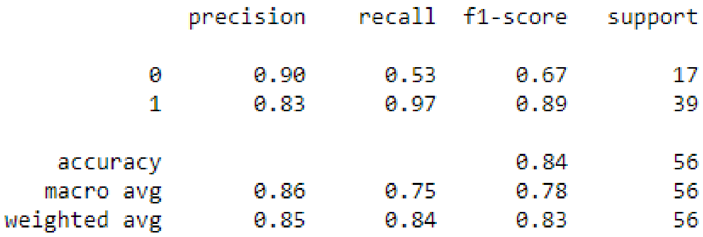


Figure 4.23: The Confusion Matrix of Naive Bayes for image

4.4.2 Audio Dataset

Similar approach like image dataset is taken for audio dataset as there the dataset is conviniently have larger value it might not give the good results as much as the others.



```
              precision    recall  f1-score   support

           0       0.90      0.53      0.67        17
           1       0.83      0.97      0.89        39

    accuracy                           0.84        56
   macro avg       0.86      0.75      0.78        56
weighted avg       0.85      0.84      0.83        56
```

Figure 4.24: The Classification Report of Matrix using Naive Bayes for Audio

4.4.3 Text Dataset

The Naive Bayes classifier is a basic classifier that is classified on the basis of event probabilities. It is also applied to the classification of texts. Since it is a simple algorithm, in many problems of text classification, it performs well. Less training time and less training data are found in other pros. It takes the probability of data and classifys the inputs according to the values which gives decent output of the dataset.

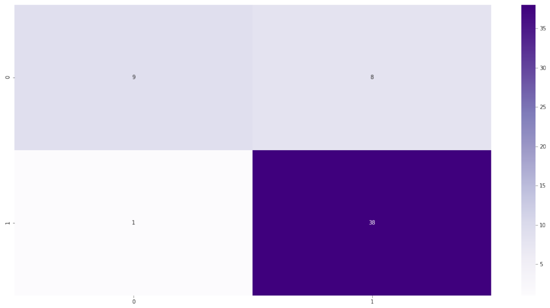
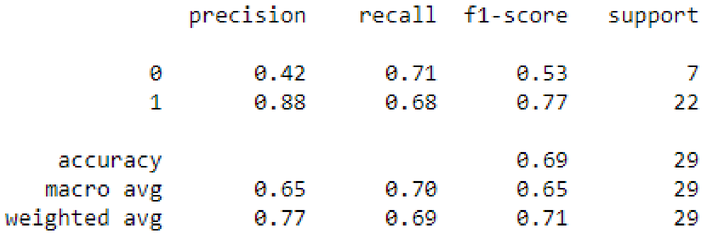


Figure 4.25: The confusion matrix of Naive Bayes for Audio



```
              precision    recall  f1-score   support

           0       0.42      0.71      0.53         7
           1       0.88      0.68      0.77        22

    accuracy                           0.69        29
   macro avg       0.65      0.70      0.65        29
weighted avg       0.77      0.69      0.71        29
```

Figure 4.26: The Classification Report of Matrix using Naive Bayes for Text

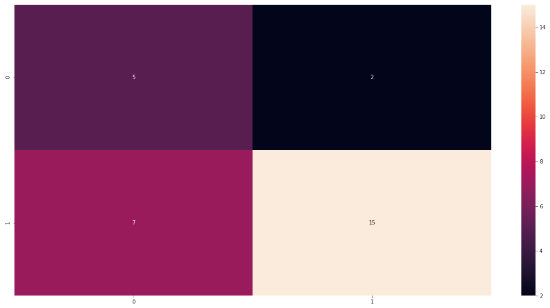
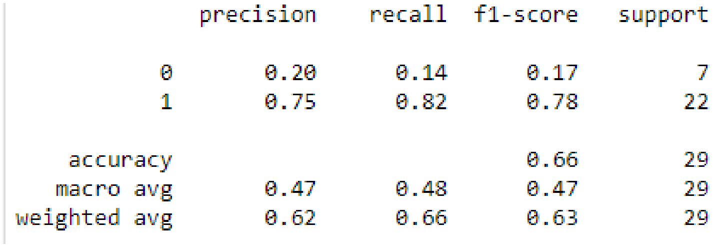


Figure 4.27: The Confusion Matrix of Naive Bayes for Text

4.4.4 Ensemble Result

After Ensemble we got 72% accuracy



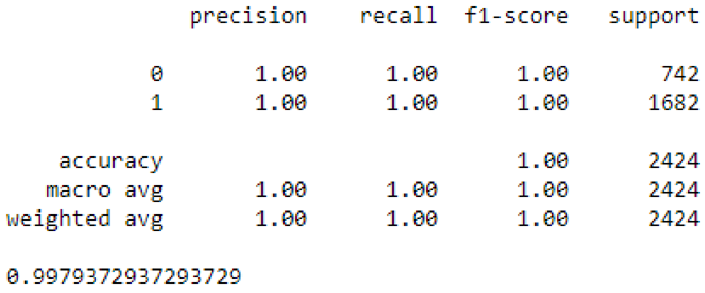
```
              precision    recall  f1-score   support

           0       0.20      0.14      0.17         7
           1       0.75      0.82      0.78        22

    accuracy                           0.66        29
   macro avg       0.47      0.48      0.47        29
weighted avg       0.62      0.66      0.63        29
```

Figure 4.28: The Classification Report of Matrix using Naive Bayes for Ensemble

4.5 Logistic Regression

4.5.1 Image Dataset

Logistic Regression is a simple machine learning algorithm that takes the image feartures to fit in the model and calculates the best results to predict the outcome. here the image features are labelled and fitted to come with the output of the image data which gives the precision for 99.6% accuray which is better than the other algorithm.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       742
           1       1.00      1.00      1.00      1682

    accuracy                           1.00      2424
   macro avg       1.00      1.00      1.00      2424
weighted avg       1.00      1.00      1.00      2424

0.9979372937293729
```

Figure 4.29: The Classification Report of Matrix using Logistic Regression for image

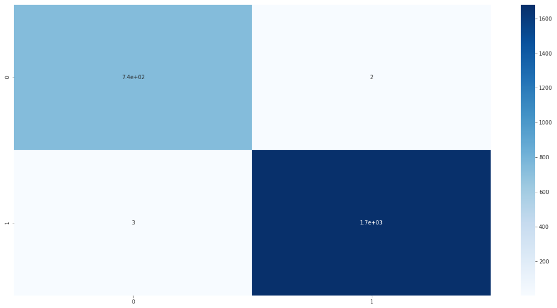
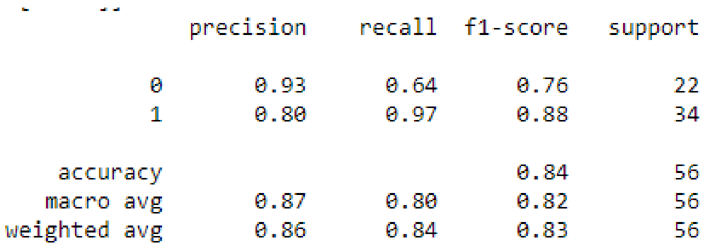


Figure 4.30: The Confusion Matrix of Logistic Regression for image

4.5.2 Audio Dataset

Similarly the audio dataset was fitted into the Logistic regression and got the outcome of precision 99.7%.



```
              precision    recall  f1-score   support

           0       0.93      0.64      0.76        22
           1       0.80      0.97      0.88        34

    accuracy                           0.84        56
   macro avg       0.87      0.80      0.82        56
weighted avg       0.86      0.84      0.83        56
```

Figure 4.31: The Classification Report of Matrix using Logistic Regression for Audio

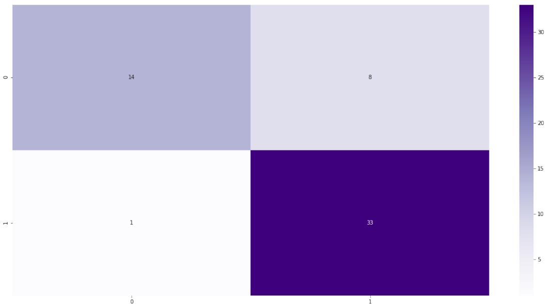
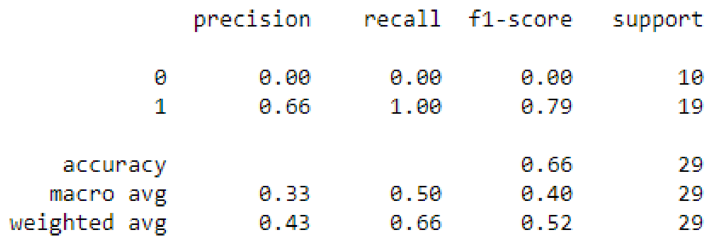


Figure 4.32: The confusion matrix of Logistic Regression for Audio

4.5.3 Text Dataset

The preprocessed dataset are fitted into the text dataset to classify the dataset which results into the precision of 70 % as the features are limited.



```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.66      1.00      0.79        19

    accuracy                           0.66        29
   macro avg       0.33      0.50      0.40        29
weighted avg       0.43      0.66      0.52        29
```

Figure 4.33: The Classification Report of Matrix using Logistic Regression for Text

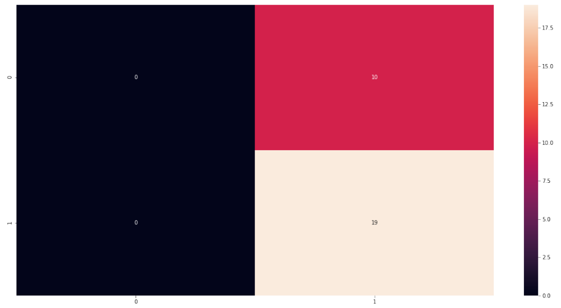
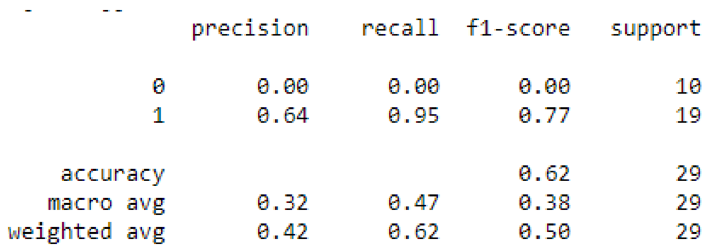


Figure 4.34: The Confusion Matrix of Logistic Regression for Text

4.5.4 Ensemble Result

After Ensemble we got 62% accuracy



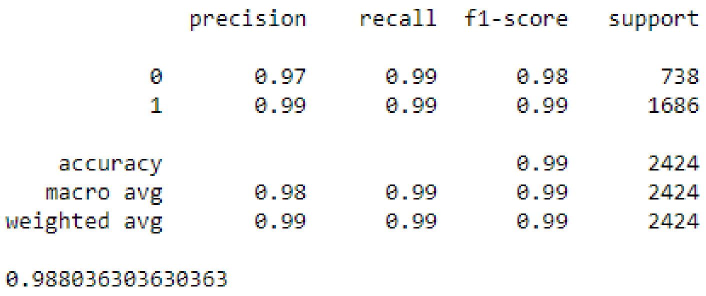
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.64      0.95      0.77        19

    accuracy                           0.62        29
   macro avg       0.32      0.47      0.38        29
weighted avg       0.42      0.62      0.50        29
```

Figure 4.35: The Classification Report of Matrix using Logistic Regression for Ensemble

4.6 Decision Tree

4.6.1 Image Dataset

Each of the images are considered of an array which the decision tree takes to identify instances by sorting them down the tree to some leaf node from the root, which provides the instance classification. Records are classified by evaluating the attribute identified by this node, beginning at the root node of the tree, then moving down the tree branch corresponding to the attribute's value Which results into two of our output hate and Non-Hate.[4] [34]



```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       738
           1       0.99      0.99      0.99      1686

    accuracy                           0.99      2424
   macro avg       0.98      0.99      0.99      2424
weighted avg       0.99      0.99      0.99      2424

0.988036303630363
```

Figure 4.36: The Classification Report of Matrix using Decision tree for image

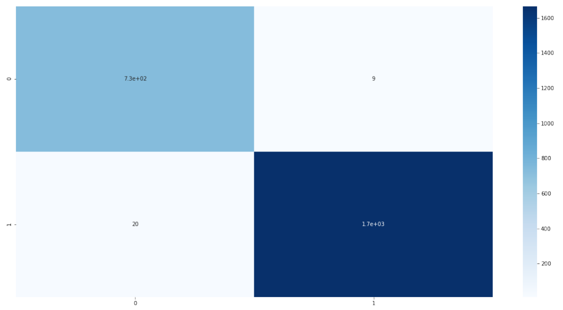
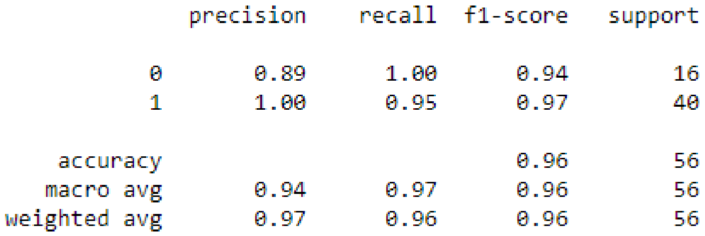


Figure 4.37: The Confusion Matrix of Decision tree for image

4.6.2 Audio Dataset

Decision tree usually performs better with the larger unbalanced data, instead of the smaller balanced sample. As the audio dataset contains matrix with large values it gives a better representation of detecting the hate and non hate classification.



```
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        16
           1       1.00      0.95      0.97        40

    accuracy                           0.96        56
   macro avg       0.94      0.97      0.96        56
weighted avg       0.97      0.96      0.96        56
```

Figure 4.38: The Classification Report of Matrix using Decision tree for Audio

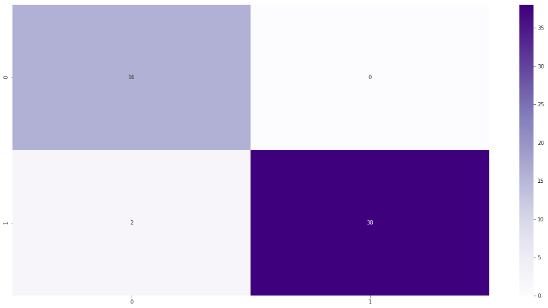
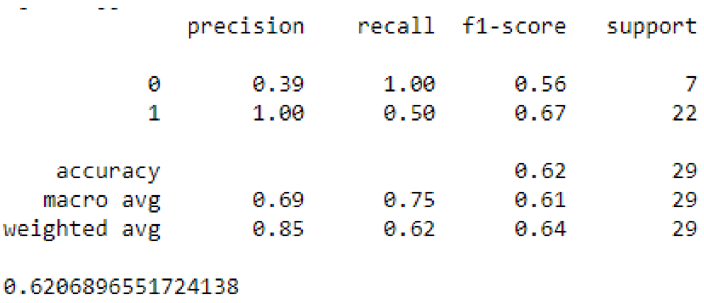


Figure 4.39: The confusion matrix of Decision tree for Audio

4.6.3 Text Dataset

The dataset displays the most common words after preprocessing and detects the hate and non-hate data from the dataset after applying Decision Tree. Decision Trees classifier,takes all the records into account to determine the features. In this matrix, all TF-IDF features are important, so the Decision Tree performs fine when predicting the outcome.



```
              precision    recall  f1-score   support

           0       0.39      1.00      0.56         7
           1       1.00      0.50      0.67        22

    accuracy                           0.62        29
   macro avg       0.69      0.75      0.61        29
weighted avg       0.85      0.62      0.64        29

0.6206896551724138
```

Figure 4.40: The Classification Report of Matrix using Decision tree for Text

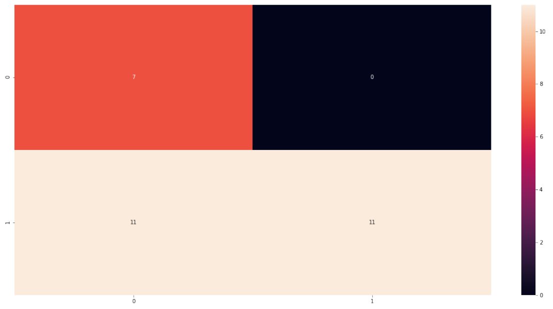
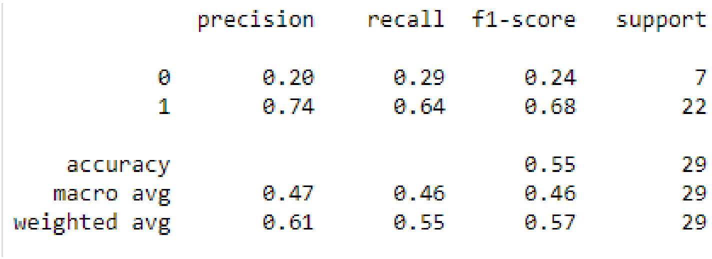


Figure 4.41: The Confusion Matrix of Decision tree for Text

4.6.4 Ensemble Result

After Ensemble we got 55% accuracy



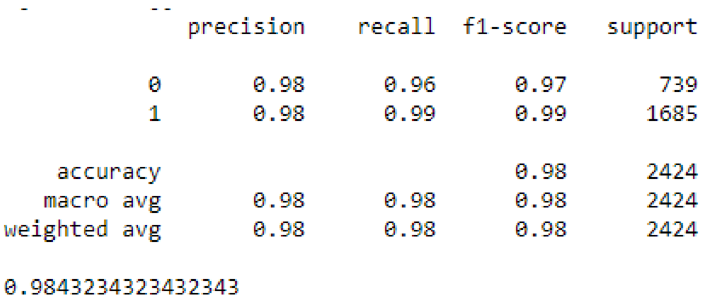
```
              precision    recall  f1-score   support

           0       0.20      0.29      0.24         7
           1       0.74      0.64      0.68        22

    accuracy                           0.55        29
   macro avg       0.47      0.46      0.46        29
weighted avg       0.61      0.55      0.57        29
```

Figure 4.42: The Classification Report of Matrix using Decision tree for Ensemble

4.7 Adaboost Classifier

4.7.1 Image Dataset

AdaBoost takes each sample of the image and correlates it with a weight that shows how important it is with regard to the classification in which all samples have equal weights. Next, it creates a decision tree with a depth of 1. for each feature. Then, to define the information that compares the predictions made by each tree with the actual labels in the training set, we use each decision tree.The next tree in the forest becomes the characteristic and corresponding tree that has done the best job of classifying the training samples. It fits well with the classifier as it takes into account the significance of the data.[7]



```
  -         --
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       739
           1       0.98      0.99      0.99      1685

    accuracy                           0.98      2424
   macro avg       0.98      0.98      0.98      2424
weighted avg       0.98      0.98      0.98      2424

0.9843234323432343
```

Figure 4.43: The Classification Report of Matrix using Adaboost Classifier for image

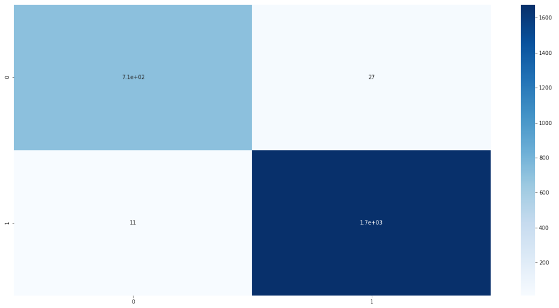
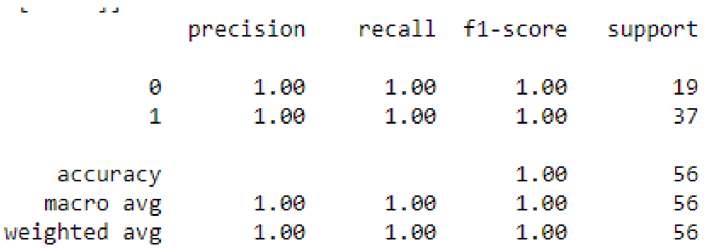


Figure 4.44: The Confusion Matrix of Adaboost Classifier for image

4.7.2 Audio Dataset

Similar procedure for audio dataset was taken to classify audio dataset which performed well with the audio featuers.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        37

    accuracy                           1.00        56
   macro avg       1.00      1.00      1.00        56
weighted avg       1.00      1.00      1.00        56
```

Figure 4.45: The Classification Report of Matrix using Adaboost Classifier for Audio

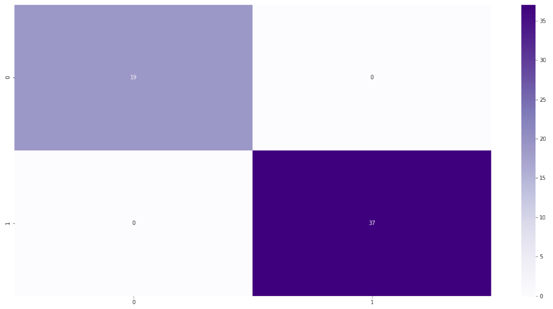
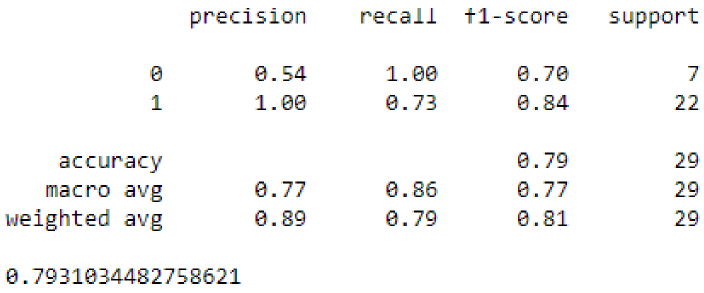


Figure 4.46: The confusion matrix of Adaboost Classifier for Audio

4.7.3 Text Dataset

Similar procedure for text dataset was taken to classify text dataset which performed well with the audio featuers.



```
              precision    recall  f1-score   support

           0       0.54      1.00      0.70         7
           1       1.00      0.73      0.84        22

    accuracy                           0.79        29
   macro avg       0.77      0.86      0.77        29
weighted avg       0.89      0.79      0.81        29

0.7931034482758621
```

Figure 4.47: The Classification Report of Matrix using Adaboost Classifier for Text

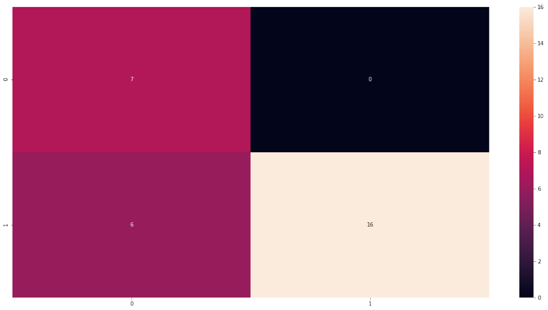
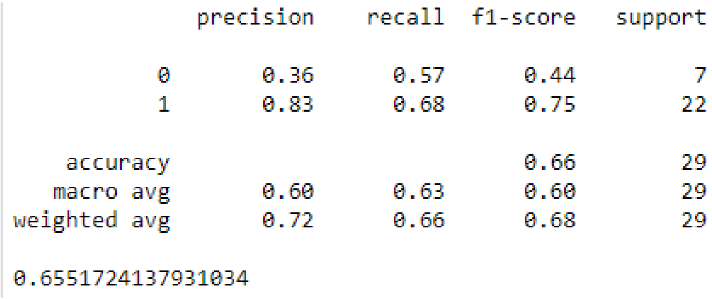


Figure 4.48: The Confusion Matrix of Adaboost Classifier for Text

4.7.4 Ensemble Result

After Ensemble we got 66% accuracy



```
              precision    recall  f1-score   support

           0       0.36      0.57      0.44         7
           1       0.83      0.68      0.75        22

    accuracy                           0.66        29
   macro avg       0.60      0.63      0.60        29
weighted avg       0.72      0.66      0.68        29

0.6551724137931034
```

Figure 4.49: The Classification Report of Matrix using Adaboost Classifier for Ensemble

4.8 Performance summary

Table 4.1: Result Analysis For Image, Audio and Text

	Data	Precision	Recall	F1Score
SVM	Image	0.9959	0.9972	0.9965
	Audio	0.9047619	0.9047619	0.91
	Text	0.875	0.913	0.893
	Ensemble	0.84	0.913	0.875
Random Forest	Image	1	1	1
	Audio	0.974	0.975	0.974
	Text	0.94117647	0.8	0.864
	Ensemble	0.75	0.9	0.818
Logistic Regression	Image	0.997	0.9976	0.9973
	Audio	0.974	1	0.987
	Text	0.740	1	0.851
	Ensemble	0.708	0.85	0.772
Adaboost	Image	0.981	0.994	0.987
	Audio	1	0.948	0.974
	Text	1	0.571	0.727
	Ensemble	0.8	0.761	0.780
k-NN	Image	0.998	0.997	0.997
	Audio	0.787	0.948	0.860
	Text	0.818	0.857	0.837
	Ensemble	0.8	0.761	0.780
Naive Bayes	Image	0.736	0.858	0.792
	Audio	0.904	1	0.95
	Text	0.687	0.7333	0.709
	Ensemble	0.565	0.866	0.684
Decision Tree	Image	0.998	0.998	0.9985
	Audio	0.926	0.904	0.915
	Text	0.875	0.913	0.893
	Ensemble	0.84	0.913	0.875

Here we can see the total performance of all the algorithm. We can see that SVM works better than the other algorithms. As we can see the Recall values differ mostly and in out experiment we need to focus on the recall.

Chapter 5

Conclusion

The process requires a large number of data to train which is one of the big limitations of the system. In future, we will apply more filters to process data. we would like to add more data to make the system more accurate. In future the system can be made as a Bot program that can remove violent or dangerous threat contents from children and detect hate speech more efficiently. We can also filter videos containing hate content or violent content. Moreover, we will apply some more classification algorithms to compare our results and build a more accurate system.Social media and many online platforms are using features like online calling and voice message sending, voice-controlled systems. As all of the popular websites are adding features to video and audio calls. It has become a new form of communication. It is only a matter of time when we will be using voice commands to operate. People from all over the world are using these features in Facebook, What's app, Skype, Zoom, Google and many more. Due to the anonymity of online platforms and social media one can easily leave hateful comments and threaten people or scam them. On the other hand, online video-sharing platforms like YouTube have a huge number of hateful and inappropriate content and comments for kids. Such deeds are causing problems such as bullying, threatening, depression, social anxiety and lots more. Many systems are present using text analysis only but it causes confusion as people use abusive words and hateful words as a sense of joke or as a part of friendly speech. Our system would detect such threat, hateful comments and inappropriate content by using image, audio and text analysis. We will be taking video as an input. We will use the image, audio and text data for feature extraction and classify them into hate and non-hate speech separately. After classifying them separately we will be combining the two results to finally determine if it is a hate speech or not. As it will be using all procedures parallely it will be fast and accurate. Moreover, it will be able to differentiate between friendly speech and hate speech with more accuracy. As the system is designed to work in any video-based content it will automatically detect any hateful comment, abusive word and threats.

In thesis defense presentation we were advised to take in more dataset to make our model more accurate an precise and was advised to take deep learning into account.

Bibliography

[1] G. M. Bhandari, R. S. Kawitkar, and M. P. Borawake, *Audio segmentation for speech recognition using segment features*, Jan. 1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-03095-1_23.

[2] R. Nishii and S. Eguchi, "Supervised image classification based on adaboost with contextual weak classifiers," vol. 2, Oct. 2004, 1467–1470 vol.2. DOI: 10.1109/IGARSS.2004.1368697.

[3] T. K. Paul, Y. Hasegawa, and H. Iba, "Classification of gene expression data by majority voting genetic programming classifier," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 2521–2528. DOI: 10.1109/CEC.2006.1688622.

[4] C. Agarwal and A. Sharma, "Image understanding using decision tree based machine learning," in *ICIMU 2011 : Proceedings of the 5th international Conference on Information Technology Multimedia*, 2011, pp. 1–8. DOI: 10.1109/ICIMU.2011.6122757.

[5] S. Arora and R. Singh, "Automatic speech recognition: A review," *International Journal of Computer Applications*, vol. 60, pp. 34–44, Dec. 2012. DOI: 10.5120/9722-4190.

[6] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the Second Workshop on Language in Social Media*, Montréal, Canada: Association for Computational Linguistics, Jun. 2012, pp. 19–26. [Online]. Available: https://www.aclweb.org/anthology/W12-2103.

[7] Y. Lee, D. K. Han, and H. Ko, *Reinforced adaboost learning for object detection with local pattern representations*, Nov. 2013. [Online]. Available: https://www.hindawi.com/journals/tswj/2013/153465/.

[8] T. Giannakopoulos and A. Pikrakis, *Introduction to audio analysis: a MATLAB approach*, 1st ed. Academic Press is an imprint of Elsevier, 2014.

[9] U. G. Patil, S. D. Shirbahadurkar, and A. N. Paithane, "Automatic speech recognition of isolated words in hindi language using mfcc," in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, 2016, pp. 433–438. DOI: 10.1109/CAST.2016.7915008.

[10] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, *Deep learning for hate speech detection in tweets*, Jun. 2017. [Online]. Available: https://arxiv.org/abs/1706.00188.

[11] Halien, *Simple image classifer with svm*, Jun. 2017. [Online]. Available: https://www.kaggle.com/halien/simple-image-classifer-with-svm.

[12] S. Polamuri, *How the random forest algorithm works in machine learning*, May 2017. [Online]. Available: https://dataaspirant.com/random-forest-algorithm-machine-learing/.

[13] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the european refugee crisis," *CoRR*, vol. abs/1701.08118, 2017. arXiv: 1701.08118. [Online]. Available: http://arxiv.org/abs/1701.08118.

[14] S. Sarkar, S. S. Solanki, and S. Chakraborty, "Time domain analysis of indian classical raga rendition," in *2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*, 2017, pp. 1–5. DOI: 10.1109/CIACT.2017.7977385.

[15] C. Bento, *Cat or dog? introduction to naive bayes*, Dec. 2018. [Online]. Available: https://towardsdatascience.com/cat-or-dog-introduction-to-naive-bayes-c507f1a6d1a8.

[16] S. Biere and P. d. S. Bhula, *Hate Speech Detection Using Natural Language Processing Techniques*, pp. 4–18, Aug. 2018. DOI: https://beta.vu.nl/nl/Images/werkstuk-biere_tcm235-893877.pdf.

[17] D. D. A. M. S. Easy, *Understanding svms': For image classification*, Aug. 2018. [Online]. Available: https://medium.com/@dataturks/understanding-svms-for-image-classification-cf4f01232700.

[18] *Five ways hate speech spreads online*, May 2018. [Online]. Available: https://edition.cnn.com/2018/05/02/us/hate-groups-using-internet/index.html.

[19] R. Gandhi, *Naive bayes classifier*, May 2018. [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

[20] R. Kshirsagar, T. Cukuvac, K. R. McKeown, and S. McGregor, "Predictive embeddings for hate speech detection on twitter," *CoRR*, vol. abs/1809.10644, 2018. arXiv: 1809.10644. [Online]. Available: http://arxiv.org/abs/1809.10644.

[21] B. Barile, *Deep nlp for hate speech detection*, Mar. 2019. [Online]. Available: https://medium.com/isiway-tech/deep-nlp-for-hate-speech-detection-25eed707997.

[22] L. Cai, Y. Hu, J. Dong, and S. Zhou, *Audio-textual emotion recognition based on improved neural networks*, Dec. 2019. [Online]. Available: https://www.hindawi.com/journals/mpe/2019/2593036/.

[23] J. Crabb, *Classifying hate speech: An overview*, May 2019. [Online]. Available: https://towardsdatascience.com/classifying-hate-speech-an-overview-d307356b9eba.

[24] O. Harrison, *Machine learning basics with the k-nearest neighbors algorithm*, Jul. 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[25] Z. Laub, *Hate speech on social media: Global comparisons*, Jun. 2019. [Online]. Available: https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons.

[26] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, *Hate speech detection: Challenges and solutions*, Aug. 2019. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371%5C%2Fjournal.pone.0221152.

[27] P. Tyagi, *Random forest*, Jan. 2019. [Online]. Available: https://medium.com/@pytyagi/random-forest-d4011bc3daca.

[28] J. Brownlee, *Boosting and adaboost for machine learning*, Aug. 2020. [Online]. Available: https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/.

[29] K. Ganesan, *Build your first text classifier in python with logistic regression*, Jul. 2020. [Online]. Available: https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python/#.YAHqa-gzaUk.

[30] *Hate crime statistics*, Nov. 2020. [Online]. Available: https://www.justice.gov/hatecrimes/hate-crime-statistics.

[31] A. Havanko, *Building a real-time hate speech detection for the web*, Oct. 2020. [Online]. Available: https://medium.com/swlh/building-a-real-time-hate-speech-detection-for-the-web-ebfb210be32c.

[32] L. Hubbard. 2020, pp. 2–27.

[33] T. S. Srivastava, *K nearest neighbor: Knn algorithm: Knn in python amp; r*, Oct. 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/.

[34] A. Angel. [Online]. Available: https://www.imageeprocessing.com/2017/07/decision-tree.html.

[35] L. Gao and R. Huang, *Detecting online hate speech using context aware models.* [Online]. Available: https://www.aclweb.org/anthology/R17-1036/.

[36] H. Shirani-Mehr, "Sms spam detection using machine learning approach," DOI: http://cs229.stanford.edu/proj2013/ShiraniMehr-SMSSpamDetectionUsingMachineLearningA
pdf.