Title: Cyberbullying Detection using Machine Learning Model

Authors: Jannat Khan (21BCY10093)
Ayush Ghogre (21BCY10063)
Prakhar Mishra (21BCY10047)
Akansha Jha (21BCY10228)

Guide: Dr. Rizwan Ur Rehman

ABSTRACT:

Cyberbullying has always been one of the most annoying issues faced by social media platforms and their users. There have always been cases of someone being disturbed due to various reasons. In this report, our main focus is on using various algorithms that can help reduce cyberbullying cases' that are faced in Indian social media space. The algorithms that we have used for our project are “Random forest classifier”,” logic regression”,” stochastic gradient descent” and “light GBM”. Now all these algorithms have certain advantages, for example, a random forest classifier can be used to analyze text and behavior of a user in real time, logic regression can be used to achieve good performance with linearly separable classes and train and set it up easily using AI model.All these algorithms in general are very helpful in dealing with abusive data available over social media sites. To use these proposed algorithms our very first step was to acquire dataset from various social media sites and train them. This dataset has been acquired by targeting the Indian audience.

Keywords: Cyberbullying, Random forest classifier, logistic regression,social media

Introduction:

Cyber bullying is defined as, using both information technology and communication technology beyond the limit in order to harm a person’s reputation, state of mind, or to humiliate a person.Cyberbullying is a problem because at any time of the day,anyone could post any hurtful comment or image targeted to a person. It is bullying in the most personalized form. A victim always remains in the fear of getting mocked or humiliated on Social Media. As kids spend most part of their lives on their mobile devices,it makes them both a better target as well as prime suspect in the realm of Cyberbullying. It is also reported “1 in 3 high school children have reported to be cyberbullied.”

Cyberbullying can take various forms:targeting someone in chat rooms, sending vulgar images/videos, impersonating someone, sharing someone’s secrets publicly or stealing passwords.

Many times, Cyberbullying comes in different structures. It doesn’t really mean hacking somebody’s profile or pretending to be another person. It likewise incorporates posting negative remarks about someone or spreading bits of hearsay to criticize somebody.

Cyberbullying affects people socially,psychologically as well as emotionally. Therefore detecting this threat is quite necessary. Detection of cyberbullying helps in identifying and classifying cyberbullying. Once cyberbullying is detected,the service providing platform can take appropriate measures.

Literature Survey:

In this section, a review of related journals and research papers is presented.Cyberbullying, real life case studies regarding cyberbullying and various approaches used in detection of cyberbullying using machine learning algorithms are discussed.

Shivashankar and Rajan(2018) described Cyberbullying as an unforeseen consequence due to the Internet which causes a deep lasting impact.It is using beyond acceptable limits or unsanctioned use of electronic technology affecting the life and reputation of another person.

It is a huge problem because a perpetrator can bully a victim at any time from anywhere.The perpetrator can be known or unknown;can be a group of individuals/or a single individual;can be continuous or one time. The paper also included real life case studies of two girls. Ritika Sharma, a school going student was stalked by a person who she has befriended on Facebook. A perpetrator has impersonated Ritu Kohli and chatted on www.mirc.com asking people to call her at odd hours on her telephone. She received 40 national/international calls in 3 days.

Venkataraghavan(2015) studied cyberbullying among teens and tweens of Chennai. For teens and tweens, social media has liberated them from social structures and parental surveillance. The use of mobile phones,applications by teens develop a sense of “Identity” in them as every feature of an application can be customized. Social media is a medium to improve one’s social status in the peer group.Cyber Bullies have certain unfair advantages such as anonymity,slow responsiveness of telecommunication services and weak law enforcement.The Journal of the American Medical Association (JAMA) reported through a fresh study in 2014 that one Indian teenager among every four is a victim of cyberbullying .The McAfee 'Tweens, Teens & Technology Report 2014’ says that 50 % of Indian tweens & teens have experienced 'cyberbullying' themselves or have seen someone else go through it.(Venkataraghavan #)
The most prevalent form of cyberbullying was found to be ‘abusive language & ‘Hurtful messages followed by ‘Mean messages’ & ‘Hurtful pictures’.

Mishra(2021) aimed to detect and classify cyberbullying using Naive-Bayes and K-Nearest Neighbor Algorithm using Chi-Square Feature Selection.Trained data was accessed from Github. The research was done on publicly available facebook content. Datasets were labeled based on severity of cyberbullying content(0(mild) to 10(severe)).An attempt was made to use this information to train the machine learning algorithms in detecting the F1 score, precision, accuracy, and recall of cyberbullying in the comments.The data was split into 70% train data and

30% test data. X_train- comment data,y_train as labeled data. A feature-selection algorithm Chi-square test used to determine the distinguishing power of each feature and also increase the performance of ml classifiers. It checks if the occurrence of specific term and occurrence of specific class are dependent. Hence, this feature selection algorithm is used to remove the features that do not correlate with the class and are not useful for classification. The evaluation metrics of this algorithm are accuracy,precision,recall and FI Score.K-NN performed better than Bayes classifier with much improvement in performance metrics and classification time.

Dr. Aarti Tolia (2016) aimed at defining Psychological effects of Cyber bullying on Children. As per the Author Bullying is deep-rooted in Indian Society since the ages through the mythological periods of Mahabharata and technology has made this issue more obvious and aggravated with new forms of bullying. Usually bullying is not considered to be a part of crime as often it is considered as a prerequisite mischief in the developing stage of children hence involvement in Cyberbullying may be as victim or the aggressor, may go unrecognized by parents, teachers and teens.

In the report the author has aimed to explore the psychological aftereffect on children and teens after being constantly harassed on social media and critically framing it as “cyber-tort”.

The paper also attempts to integrate existing dilemmas due to cyber-bullying, and omnipresent personality traits in an online environment and deduce it to the psychological context.

As per the National Center of Educational Statistics one out of three students is bullied during the school year, another survey conducted by Sameer Hinduja and Justin W. Patchin (2015) states that 34% students experienced cyber bullying from the total number of children surveyed. It was also observed that adolescent girls are significantly more likely victims of cyberbullying in their lifetimes (40.6% vs. 28.2%)

- **Psychological Effect:**

Psychological effects of cyberbullying may not be noticed initially but the negative stress factor builds up overtime. There is an increased risk of developing a number of health and psychological problems because of negative socializing through social networking. The after effects of Cyberbullying on social networking sites may create emotional distress after or during the use of the internet. The child may also cut off himself from gathering and might stop socializing with friends and family. The victim often skips school with considerable change in mood, sleep and appetite, the grades may fall and the child may become aggressive for no apparent reason.

Few other after effects might be anxiety, depression and stress related disorders, academic difficulties, alcohol or drug abuse without the knowledge of parents, attention and retention problems, bad dreams and bed wetting, chronic pain, dangerous behavior such as speeding, eating disorders, failure to thrive, fear or shyness of certain adults or places.

Ms. Surabhi Negi (2016) studied cyberbullying and came to know all these facts. Teenagers not only use technology to communicate, but also as a way of developing their identity. The popularity of these new adolescent communication tools has created some new challenges as well as some negative adolescent behavior. The effects of cyberbullying are more damaging than traditional bullying and have longer lasting effects. The effects can be social, emotional and academic maladjustment of the victims, which will be identified by the research. The exploratory design is a two-phased mixed methods design that uses quantitative data to explain qualitative results (Creswell & Plano Clark, 2007). The researcher has utilized both qualitative and quantitative methods. The sample for the focus group discussion comprised participants who had previously reported the incidents of cyber bullying with the school counselor. The sample consisted of four girls and three boys studying in classes IX and X. Lottery system was used for selecting the sample of 10 students for conducting the interview. The participants gave their consent for being part of the study. Stratified random sampling technique was utilized to choose a sample of 120 students of classes IX and X for the administration of the questionnaire. findings of the questionnaire administered showed that almost two-thirds of participants [n=81] understood the meaning of the term cyber bullying in its true context. Through the questionnaire the respondents were asked to enlighten which gender as per them is at a higher risk when we talk about cyber bullying and more than half of the respondents 52% [n=62] felt that both genders are at equal risk when it comes to cyber bullying. When the question of being cyber bullied was posed to a larger sample through the questionnaire, 45% [n-=53] of the respondents revealed that they have occasionally witnessed incidents of cyber bullying while these episodes were often noticed by 36% [n=42] of the respondents.

Adya Bansal, Akash Baliyan, Akash Yadav, Aman Kamlesh, Hemant Kumar Baranwal (2022) used AI and ML methods to calculate some facts on cyberbullying. The aim of this project is to classify a set of data as cyberbullying or not and to compare five machine learning algorithms and find the most accurate one for the classification. The ML algorithms used were Naive Bayes, Decision Tree, Random Forest, SVM, DNN Model using Python. In this research they used five machine learning algorithms and studied them carefully to find out the algorithm with best accuracy among all five of them. To find out which algorithm is best they trained each one of them on the same dataset and compared them with each other more precisely. The dataset used for this study is downloaded from a website called kaggle.com. The dataset contains two types of sets which are bullying text and non-bullying text. The goal is to identify all the bullying text. Keeping in mind the importance of a system which can detect cyberbullying and online harassment, they studied different ML algorithms and their effectiveness in comparison with each other to predict their accuracy on a given data set to find out the best among them. After studying all the five algorithms and their results they came to a conclusion that the DNN model performs best in detecting cyberbullying with an accuracy of 0.990145 and along with the second-best performing algorithm comes out to be random forest algorithm with an accuracy of 0.986897.

Blaya, Kaur, Sandhu (2018) in their study noticed that There is a dearth of research on the potential risks associated with the use of electronic media. It can be said that research in India on issues like online behaviors of children and youth, cyber bullying, and various other negative online experiences is still in its infancy. Recently, Indian researchers from various fields, especially psychology, information technology, and criminology, have started to explore youth access, usage, and negative online experiences. A study found that 45 % of Indian parents believed that a child they knew was being cyberbullied, while 53 % were aware of the issue of cyberbullying. Three out of 10 parents in India say that their children have been victims of cyberbullying; this was most often through informal communication destinations like Facebook (60 %) Globally, call phones and online talk rooms were a far - off second and third, each around 40 %; in India, it is equal between long - range interpersonal communication locales like social networking sites (55 %) and online talk rooms (54 %). India was found to have the third highest rate of online bullying . Among the Indian young people who responded , 53 % had experienced cyberbullying and 50 % had bullied someone online.

Nalwa and Anand (2003) investigated the extent of Internet addiction in a small sample of schoolchildren (16-18 years old) in Punjab. Two groups were identified, dependents (18) and non - dependents (21). Dependents, who spent more time on the Internet, were found to delay other work to spend time online, lose sleep due to late - night logons, and feel life would be boring without the Internet. The dependents also scored higher than the non - dependents on a loneliness measure. Internet addiction and its correlates were explored among high school students in years 11 and 12 from Ahmedabad, India by Yadav and colleagues (2013). Out of the total sample of 621 pupils, 65 (11.8 %) had Internet addiction. The most common online activity was found to be social networking (98 %), followed closely by emailing (88 %). The prevalence of moderate to severe Internet addiction appeared to be low, with 17 students (19 %) classified this way. Vidyachathoth and colleagues (2014) explored the association between addiction to the Internet and individuals ' affect. This cross - sectional study involved 90 individuals from Mangalore, in the south of India, aged 18-20. Participants were selected randomly from the first year of medical undergraduate students. They found a significant positive correlation between Internet addiction and negative affect. Daily duration of Internet use and negative affect also had a positive correlation, highlighting the role of affect in behavioral addictions .

Rounak Ghosh, Siddhartha Nowal and Dr. G. Manju aimed to build a ML model which can detect bullying text in Bangla language using various algorithms such as Support Vector Machine (SVM), Logistic Regression (LR), Random Forest and Passive Aggressive (PR) classifier. The research was done on publicly available comments from Facebook and Twitter. The complete data was preprocessed by the steps such as data cleaning, stop word filter, transforming whole data into lower case, tokenization to facilitate feature extraction.

Sheeba, devaneyan, and caravane (2019) came up together to analyze and study the impact of bystanders' direct intervention in cases of cyber harassment. They aim to identify and classify cyberbullying incidents using the bystander intervention model. Bystanders can play a positive and active role in defending a cyberbully victim by reporting bullying incidents to higher authorities or by providing relief to victims. “According to the census of cyberbullying in 2014, 50% of the youngsters report that they have accomplished repeated bullying through mobile phones or on the internet.” It was also observed that about 80% of recurring cases of cyberbullying occurred through facebook.Hence, the authors of this research paper proposed a work that focuses on the impact of the direct intervention of a bystander. The researchers adopted latent semantic analysis(LSA) for detecting signs of cyberbullying in the comments section of Facebook from shorthand text and emoticons and a Random Decision Forest algorithm for classifying words used for cyberbullying. This would basically work to send an alert message to a cyber-victim and a sort of warning to the bully to stop their actions, it would also be capable of differentiating between a bullying message from normal one and would allow a bystander to block or report a bully with the victim's permission.

Pavitar, Vijay and Majid(2019) observed that most of the materials available for cyberbullying on the internet are of foreign origin. Even after being home to a large population there is very small content available. That’s why they came up with “Cyber Bullying as an Outcome of Social Media Usage: A Literature Review”. This work is a review of all the existing literature on the topic of cyberbullying in India.In this literature review, the authors review all the research papers or reports available on cyberbullying in India. During their study they found that with the rise in use of technology there has been a rise in time being spent by Indian users over the internet making them more vulnerable to cyberbullying. It was also observed during their study that due to lack of awareness and poor implementation of cyber laws in India, conditions are more severe.In their review the authors also give information regarding the difference between cyberbullying and traditional bullying. They discuss these differences on the grounds of location of being bullied, amount of time that bullying is being faced, different forms or ways of being bullied. A sense of powerlessness can be observed from the victims who are facing cyberbullying through these small differences. In traditional bullying a victim can predict time or location where they are going to be bullied but in case of cyberbullying even this small information remains unpredictable.

Shilpi and Soni(2019) studied cyberbullying together and came up with “Perceived vulnerability of cyberbullying on social networking sites: effects of security measures, addiction and self-disclosure”. This paper mainly focuses on finding out the reason why even after so many years of effort cases regarding cyberbullying keep on increasing. During their study, they found that with improvement in security measures being used by social media sites, a surge of lax behavior has been observed from the user. Where they voluntarily give their personal information which leads to addiction in use of social media which further leads to exposure of cyberbullying. They also observed a huge gender difference in the awareness of security measures, addiction intensities, level of self-disclosures and propensity to cyberbullying victimization. To deal with this, they proposed a research groundwork in (TTAT) Technology Threat Avoidance Theory.

Research Gap:

1. **The anonymity of the perpetrator:**

   When a perpetrator bullies a victim on social media, they can be anyone from their classmate to cousin to next door neighbor to an unknown person.

   But on social media, people can be anonymous.

   Hence there is no way for the victim to know who has bullied them.

   The existing measures to know the perpetrator are pretty limited and not useful.

2. **Existing measures to detect cyberbullying are not sufficient:**

   The existing models to detect cyberbullying are not enough. Their implementation is quite rare. They are not as effective as they should be. They even didn’t accurately predict the cyberbullying scripts. Hence, we need a practical model to implement meticulous detection of cyberbullying scripts.

3. **Enough datasets for regional languages are not available to train a single ML Model**

   There are a few Machine Learning models to detect Cyberbullying text, but most of them are for English language. There is no such model which could detect Cyberbullying text in any of the regional indian languages and the main reason for it could be the scarcity of data, there is almost no data available to train and test the ML model. Also there is no such ML model which has the capability to detect cyberbullying text in all of the regional

Indian languages, like a ML model which can detect cyberbullying text in Hindi as well as Marathi, Bengali, etc.

One of the drawbacks of the existing ML model is that sometimes people try to communicate in one language but type it in another language. For Example, in our day to day whatsapp communication what we want to communicate is in hindi, but we type it in english. So in that case ML model fails to detect Cyber bullying.

4. **Bystander negative impact**

   There are certain cases observed where due to intervention of bystander, conditions of a cyberbullying victim have only detroited. The existing research papers only talk about the positive impact of bystanders in cyberbullying. But we don’t have data regarding what will be the negative aspect of bystander intervention in cyberbullying.

5. **Increase in giving personal information social media**

   A certain lax behavior has been observed by people as they very easily provide their essential information over social media blindly. With increase in security there has been an increase in this lax behavior by users. But we don’t have any data regarding how to deal with this situation.

6. **IP address information on social media**

   An IP address indicates what city you’re in. Once someone knows that, they may also be able to poke around online and find your actual address. There are no clear ML detection techniques to identify if someone has leaked your IP address in a comment chain. Sometimes during a heated online argument one person may reach some extreme levels of online bullying. This can result in the bully to spam you and get a hold of your IP address which can cause harm even to your safety.

If leaked online, many home invaders keep an eye on social media so they know when homeowners go out of town. If they have IP addresses, it’s usually pretty easy to know which houses to hit.

Your geolocation isn’t just important to potential thieves. With your IP address, someone can execute a DoS (denial of service) attack against you. Bullies can also use your IP address to frame you for crimes that you didn't commit.

Cyberbullying at such heights can also result in activities such as stalking as not everyone wants to use your IP address for illegal purposes.

The worst case scenario that can happen is that the bully sells your IP address on the dark web. Not all thieves are looking to cash in by impersonating you. A cyberbullying detection system also should be able to detect any use of numerical and decimal values in the text that can be helpful to raise suspicions on the leaks of IP addresses.

Proposed Model

- □ ***Main Diagrams***

1. Implementation of Our Model:

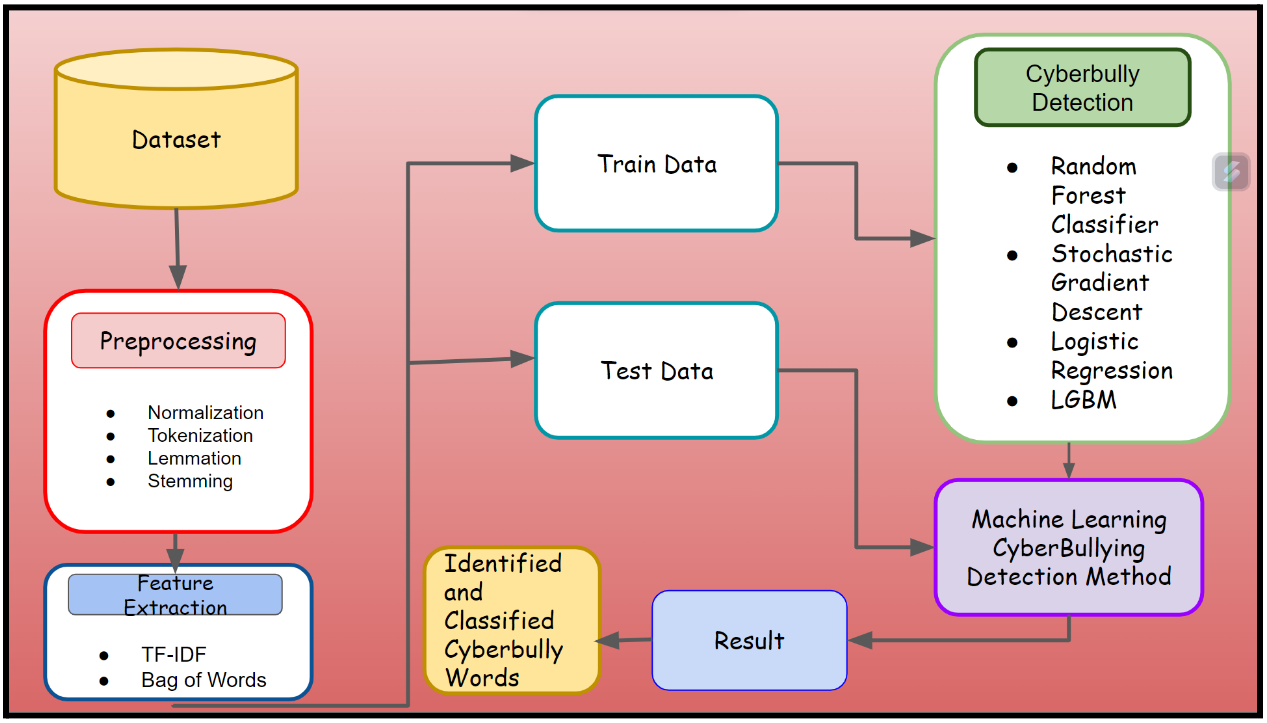


Figure 1.

Here figure 1 represents the main implementation of our Machine Learning Model.

- **Acquiring Datasets**
  - ★ The cyberbullying detection model we are building is targeted for the Indian audience.
  - ★ Here we have acquired datasets which included the comments made by Indian audiences to Indian Social Media Influencers, Indian Politicians, Bollywood Celebrities, TV Stars.
  - ★ The cyberbullying comments were obtained from Twitter, Instagram and Youtube.

  - The scripts were classified into numerical categories. ‘0’ is considered as non-cyberbullying and ‘1’ is considered as cyberbullying.
  - We obtained 300 scripts and classified them as cyberbullying or non-cyberbullying.
  - The link to our dataset is: https://docs.google.com/spreadsheets/d/1qr4BdfW2wGRGHdO2e-cx0XaLaVOjeTv_PWbpOwDAM-o/edit?usp=sharing
  - The dataset is structured.

- **Data Preprocessing**
  - The next step of building our ML Model is “Data Preprocessing”.
  - The first step in Data Preprocessing is Normalization. Normalization is done in order to reduce higher variance in data. Normalization brings all the numerical data between 0-1. Our data is already in the form of 0/1, hence our data is already normalized.
  - Our next step is to Tokenize the dataset. Our script is to split the script into sentences and then the sentences into words. As our model will detect cyberbullying script on the basis of specific keywords, it is necessary to tokenize our sentences into words.
  - Our final step in data preprocessing is Lemmatization and Stemming. Stemming reduces the word to word stem. For example, ‘eating’ is changed to ‘eat’. Stemming makes our dataset easier to process. It helps in utilizing space effectively. Lemmatization reduces the words to dictionary format. Lemmatization can be considered as an intelligent form of Stemming.
- **Feature Extraction**

  The third step in building our ML Model is to do “Feature Extraction”.

We have used 2 NLP Techniques here.

1. Bag of Words: Let us take 2 sentences

    a. The sun is shining.

    b. The rain is pouring.

    Here there are 6 unique words as ‘the’ and ‘is’ is repeated 2 times. BoW is a technique where we replace the words with their frequency.

Words	Frequency
the	2
sun	1
is	2
shining	1
pouring	1

2. Tf-IDF: In this NLP technique, the words in abundance which are less significant (like, and, this, or) are given less weightage whereas rare words are given more weightage.

Here Tf is Term frequency. The formula for Tf is:

$$Term\ frequency\ of\ a\ word\ ‘x’ = \frac{count\ of\ ‘x’\ in\ the\ total\ words}{total\ no\ of\ words}$$

Df is total no of documents that hold ‘x’.

N is total no of documents

$$idF = log(\frac{N}{dF})$$

- **Train and Test Data**

  Now we split our preprocessed data into Train and Test. Train data is sent to machine learning algorithms. Once the model is tuned, the Test data is used to validate the results. Once the model is tested and validated it can be used to identify Cyberbullying scripts.

2. User-Model Interaction:

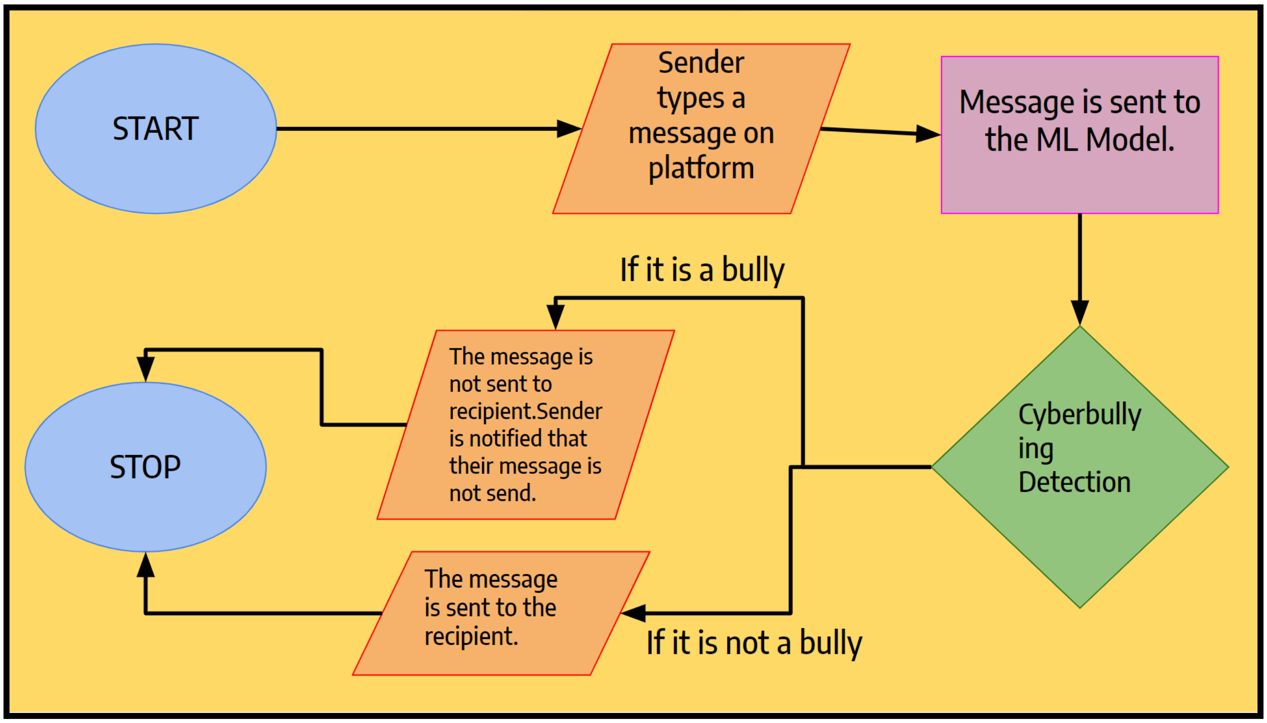

START
Sender types a message on platform
Message is sent to the ML Model.
If it is a bully
The message is not sent to recipient.Sender is notified that their message is not send.
STOP
Cyberbully ing Detection
The message is sent to the recipient.
If it is not a bully

Figure 2.

Figure 2. demonstrates the user's interaction with the ML model. We are deploying this model on discussing forums, chat-rooms, Twitter-threads, Instagram-comments, Youtube-comment section and even our personal chats on Social media platforms. The user sends a message on the mentioned platform. The message is then sent to the model. The model detects whether the message contains cyberbullying words. If the message contains cyberbullying words, then the message is not sent to the recipient and the user is notified that the message is unsent as it contains harmful content. If the message doesn't contain any cyberbullying words, then the message is sent to the recipient. In this way, an user interacts with the model.

☐ ***Sub-Diagram***

- Data Preprocessing:

START

↓

We have a dataframe divided into columns, script and label.

Script	Label
I believe that raining will solve problems.	0

↓

1.Normalization:
As the label is always 0/1, there is no need for normalization. The data is already normalized.

↓

2. Tokenization:

Script	Tokenized script
I believe that raining will solve problems.	{‘I’,’believe’,’that’,’raining’,’will’,’solve’,’problems’}

↓

3. Lemmatization and Stemming:

Script	Lemmatized and Stemmed script
I believe that raining will solve problems.	{‘I’,’believe’,’that’,’rain’,’will’,’solve’,’problem’}

↓

END

Algorithms:

1. ***Random Forest Classifier:***

It is a supervised Learning algorithm which is used for regression and classification problems. The decision made by this algorithm is based on random selection of data samples and getting predictions from every tree.

2. ***Logistic Regression:***

Logistic regression is a supervised classification machine learning model.

In our dataset, the label has only two values 0/1. Hence our logistic regression model is binomial.

Step 1: Obtain the sigmoid curve equation

Mathematical derivation of Sigmoid Curve for logistic regression model:

$$Order\ of\ Success\ =\ Odd(\theta)\ =\ \frac{Probability\ of\ event\ happening}{Probability\ of\ event\ not\ happening}$$

$$\theta \;=\; \frac{p}{1-p}\;,\;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(i)$$

where θ is Success of an event

p is the probability of an event happening.

Here θ varies as follows $0 \;\leq\; \theta \;\leq\; \infty$

As our labeled dataset has discrete values, we have to convert the line into a sigmoid curve so that logistic regression can be applied to the dataset.

Conversion of linear equation to Sigmoid Curve equation:

Equation of line: $y = B_o + B_1x$ ……………………………(ii)

Combining (i) & (ii) and taking log of (ii)

$\Rightarrow$ $\log(\frac{p(x)}{1-p(x)}) = B_o + B_1x$ …………………………………(iii)

$\Rightarrow$ $\frac{p(x)}{1-p(x)} = e^{Bo + B1x}$ ……………………………………..(iv)

Substituting $e^{Bo + B1x} = Y$ in (iv), we get

$p(x) = Y(1 - p(x))$ ……………………………………………(v)

$\Rightarrow$ $p(x)(1+Y) = Y$ ……………………………………………..(vi)

$\Rightarrow$ $p(x) = \frac{Y}{1+Y}$ ……………………………………………….(vii)

Substituting $e^{Bo + B1x} = Y$ in (vii), we get

$\Rightarrow$ $p(x) = \frac{e^{*}Bo\,+\,B1x}{1+e^{*}Bo\,+\,B1x}$ ………………………………………(viii)

The equation (viii) is an equation for the sigmoid curve.

Step 2. Set a threshold value ‘x’.

Once the sigmoid curve is plotted, a threshold value is set.

Step 3. Predicting the unlabelled data on the basis of threshold value.

If the value of input data in the sigmoid equation is less than the threshold value, then it tends to 0 and if its value is more than threshold value then it tends to 1. In this way the unlabelled data is labeled to 0/1.

3. ***Light GBM:***

<u>Step 1:</u> Employ Smart Feature Extraction Techniques:

- Smart Feature Extraction Techniques ------> Boost the algorithm

<u>1. Exclusive Feature Bundling:</u>

- Exclusive Feature Bundling is one of the Smart Feature Extraction Techniques.
- For example, male and female is an exclusive feature i.e. if a person is male,they cannot be female and vice versa.

Male	Female
1	0
0	1
1	0
0	1

- The male and female categories can be bundled together to form one single category.
- Here 10-represents male, 11-represents female.

Gender
10
11
10

11

- This technique reduces the dimensionality of the dataset.

2. Binning Method:

- It is a feature extraction technique which increases the efficiency of the algorithm.
- Here is how it works: Suppose we have given the below dataset.

Instances	Frequency
10	2
23	3
22	1
12	3
11	3
31	4
24	2
32	6
27	7

- The instances can be binned together as follows:

Interval	Frequency
10-20	8
20-30	13
30-40	10

Step 2: Boosting Algorithm:

- Ensembled techniques are used in the boosting algorithm.
- Decision Tree is employed everytime sampling is done.

Step 3: GOSS(Gradient based One Side Sampling):

1. We take a baseline model M_0.
2. Decision tree algorithm is applied on it.
3. Gradients are obtained G_0,G_1,..........G_N.
4. Above gradients are arranged in descending order.
5. Top 20% of this gradient is put in a sample.
6. Of the remaining 80% gradients, a random 10% gradients are chosen and put in the same sample as in 5.
7. If in the top 20%, the gradient is low, it means the model is performing well and there is no need to train it.
8. However, if the gradient is high in 20%, the model should be trained again on the sample obtained in 6.
9. The iteration in 8 continues till the gradient is low in the top 20%.
10. In this way, a final M_N model is obtained.

4. Stochastic Gradient Descent (SGD):

Gradient descent is an iterative algorithm that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function. The formula for Stochastic gradient Descent is:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

The steps of the algorithm are:

- Choose a starting point (initialisation).
- Calculate gradient at this point.
- Make a scaled step in the opposite direction to the gradient.
- Repeat points 2 and 3 until one of the criteria is met.
- Maximum number of iterations reached.
- Step size is smaller than the tolerance.

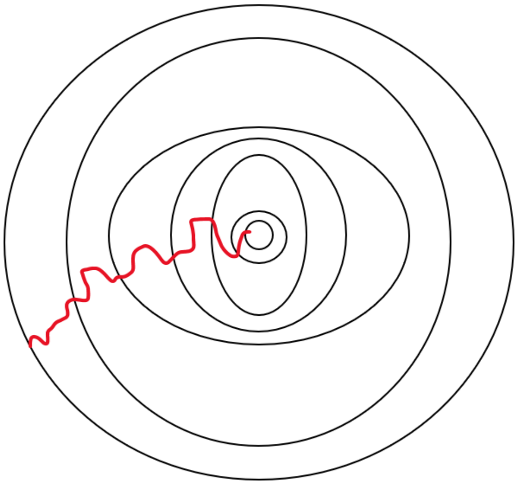
SGD Algorithm →

For *i* in range (m):

$$\theta_j = \theta_j - \alpha\,(\widehat{y^i} - y^i)\,X_j^i$$

In SGD, only one sample from the dataset is chosen at random for each iteration, the path taken by the algorithm to reach the minima is usually noisier than your typical Gradient Descent algorithm. But that doesn’t matter all that much because the path taken by the algorithm does not matter, as long as we reach the minima and with a significantly shorter training time.

Path taken by Stochastic Gradient Descent –



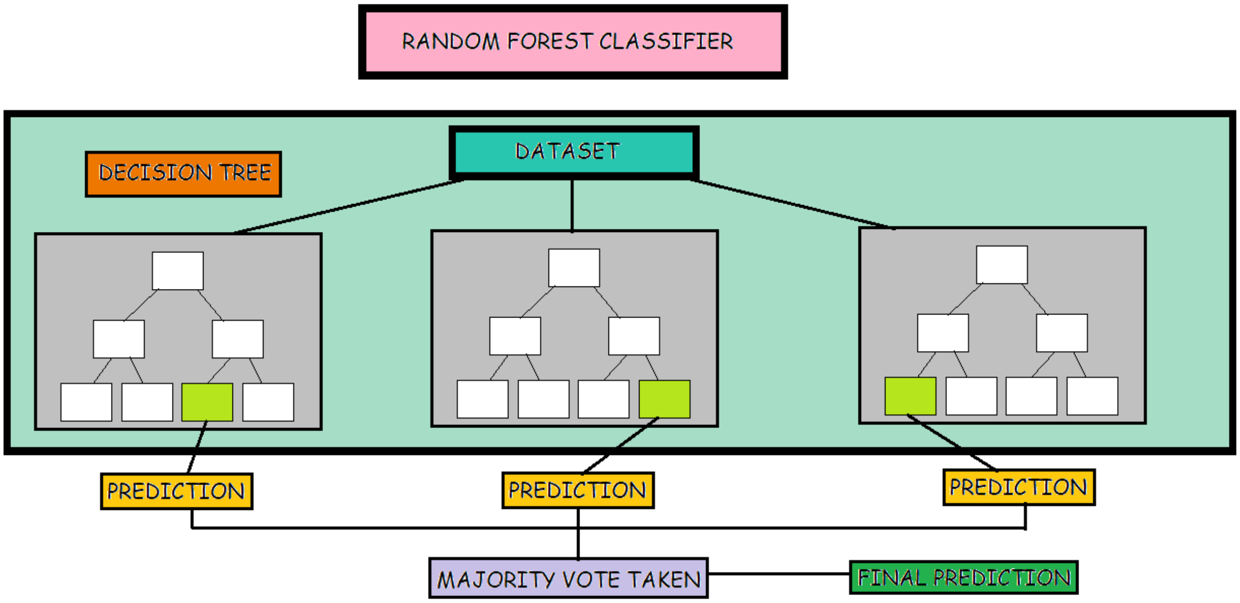
Advantages of Stochastic Gradient Descent:

- It is easier to allocate in desired memory.
- It is relatively faster to compute than batch gradient descent.
- It is more efficient for large datasets.

Flowcharts:

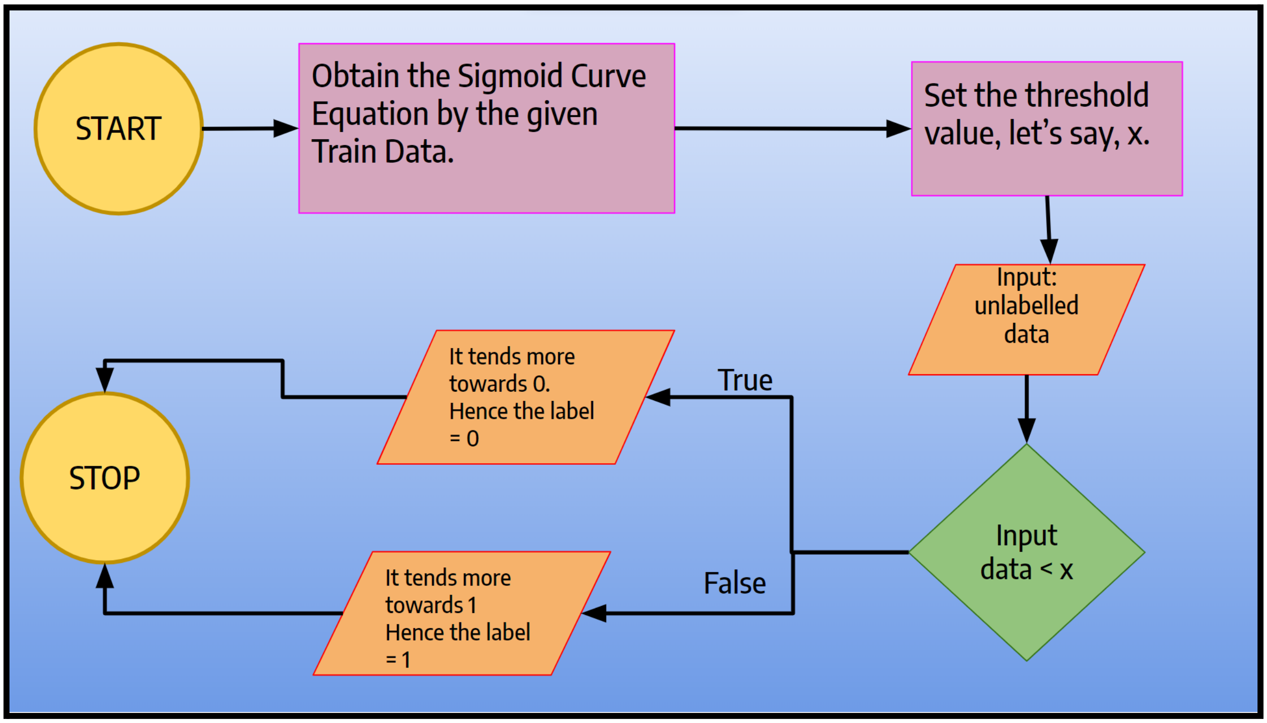
1. Random Forest Classifier:

   Here is the flowchart of the working of Random Forest Classifier.



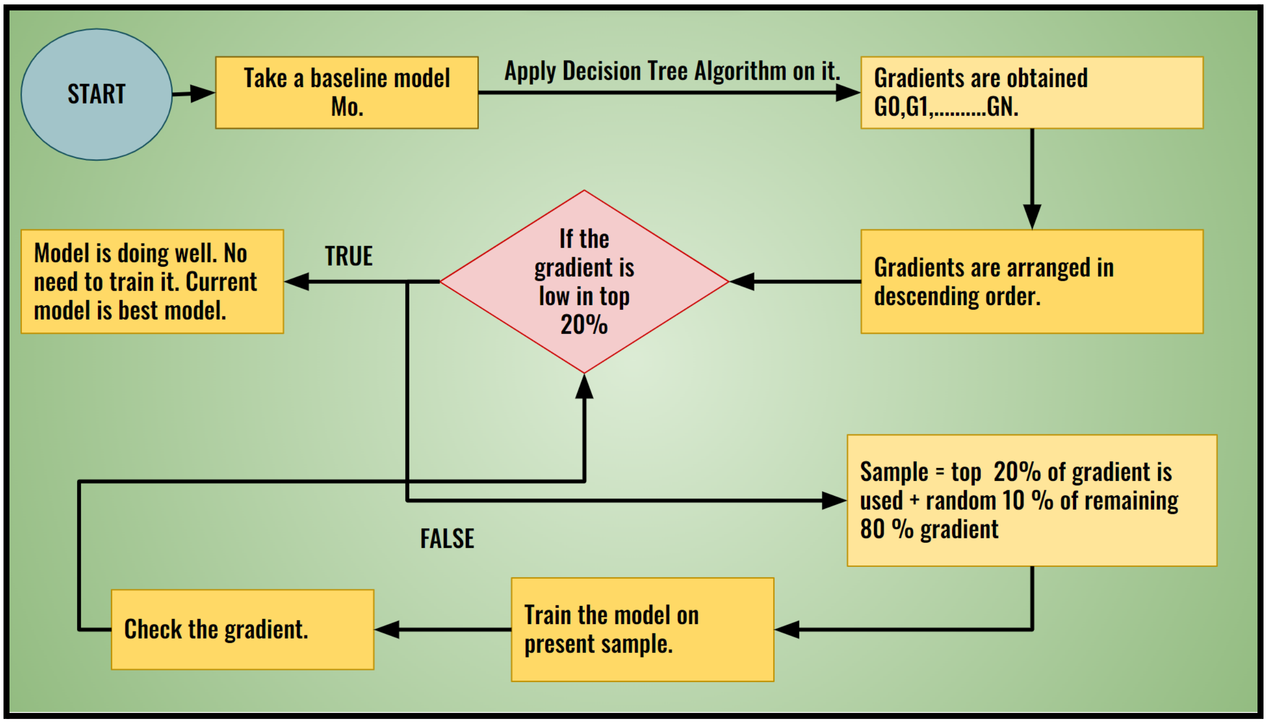
2. Logistic Regression:

Here is the flowchart of the working of Logistic Regression model.


START
Obtain the Sigmoid Curve Equation by the given Train Data.
Set the threshold value, let’s say, x.
Input: unlabelled data
Input data < x
True
It tends more towards 0. Hence the label = 0
False
It tends more towards 1 Hence the label = 1
STOP

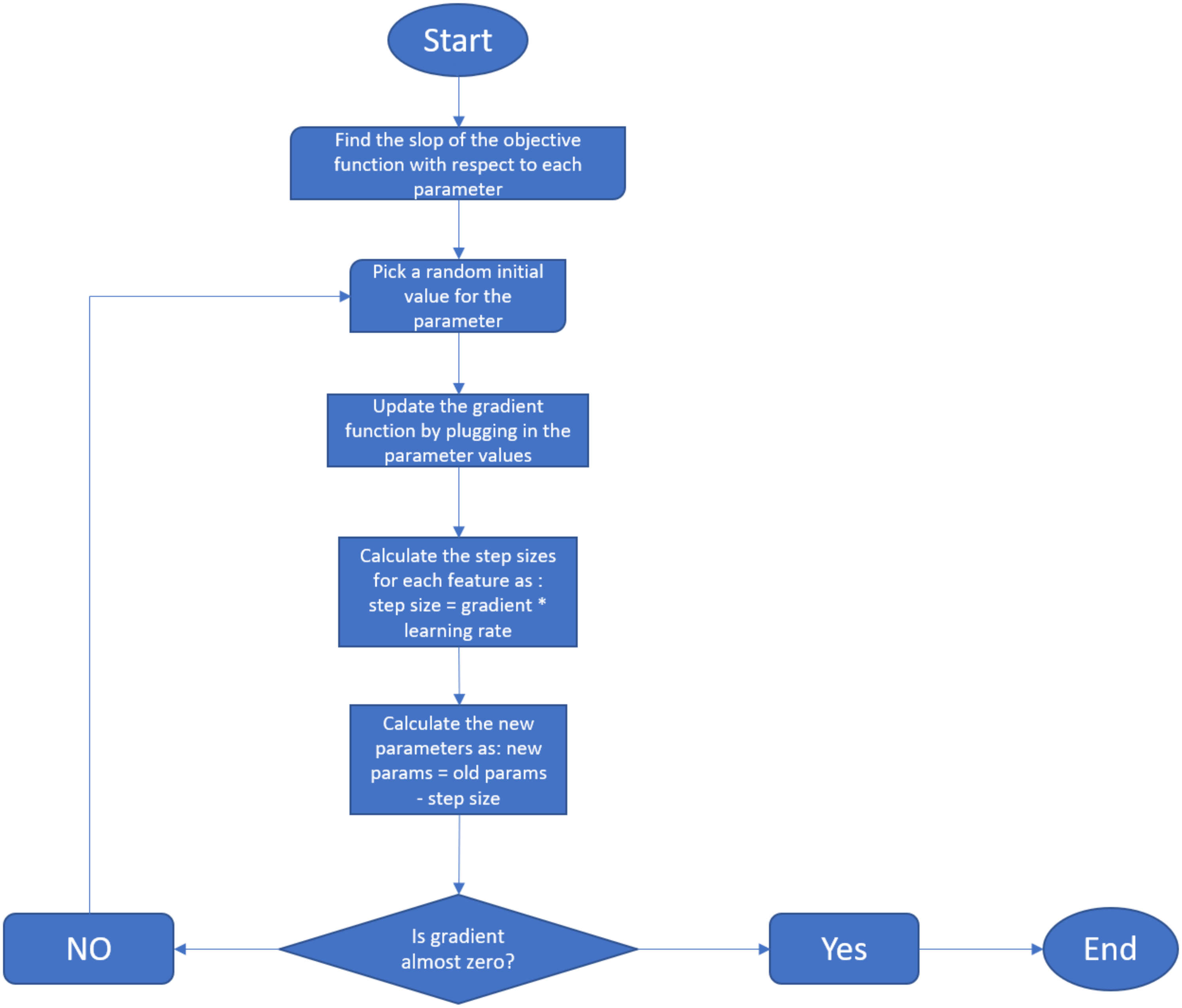
3. <u>LightGBM:</u>

Here is the flowchart of Gradient One Side Scaling of LightGBM.


START
Take a baseline model Mo.
Apply Decision Tree Algorithm on it.
Gradients are obtained G0,G1,..........GN.
Gradients are arranged in descending order.
If the gradient is low in top 20%
TRUE
Model is doing well. No need to train it. Current model is best model.
FALSE
Sample = top 20% of gradient is used + random 10 % of remaining 80 % gradient
Train the model on present sample.
Check the gradient.

4. <u>Stochastic Gradient Descent:</u>

Here is the flowchart of the working of SGD model.


Start
Find the slop of the objective function with respect to each parameter
Pick a random initial value for the parameter
Update the gradient function by plugging in the parameter values
Calculate the step sizes for each feature as : step size = gradient * learning rate
Calculate the new parameters as: new params = old params - step size
Is gradient almost zero?
NO
Yes
End

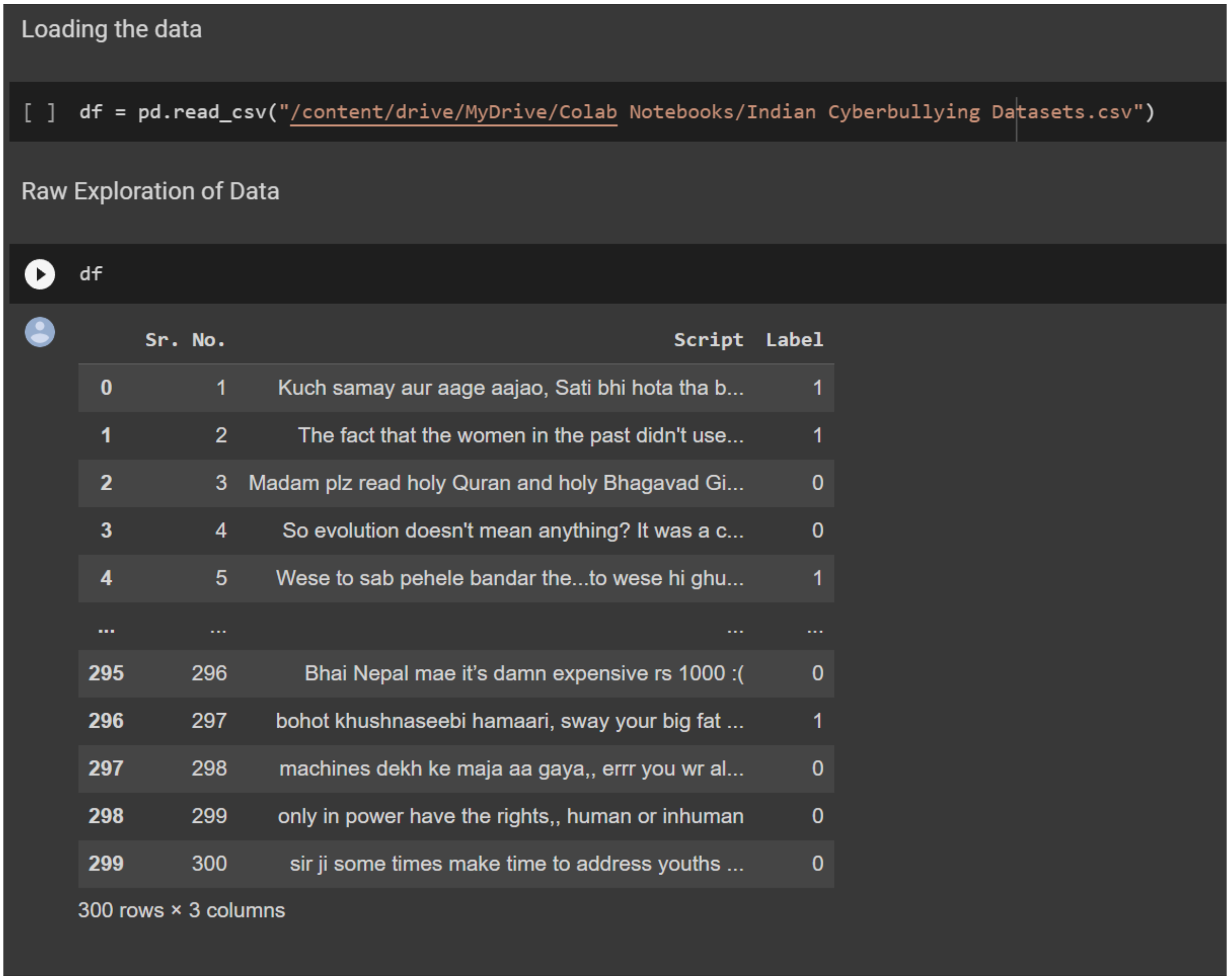
Code Section:

1. Software Requirements:
    - Google Colaboratory
2. Language used:
    - Python
3. Libraries used:

- pandas
- numpy
- nltk
- stopwords
- nltk.tokenize
- word_tokenize
- WordNetLemmatizer
- words
- TfidfVectorizer
- train_test_split
- matplotlib.pyplot
- seaborn
- plotly.express
- plot_confusion_matrix, classification_report
- import SGDClassifier
- import LogisticRegression
- RandomForestClassifier
- enable_halving_search_cv
- HalvingGridSearchCV
- lightgbm

4. Code Snippets

I. Exploratory Data Analysis



Loading the data

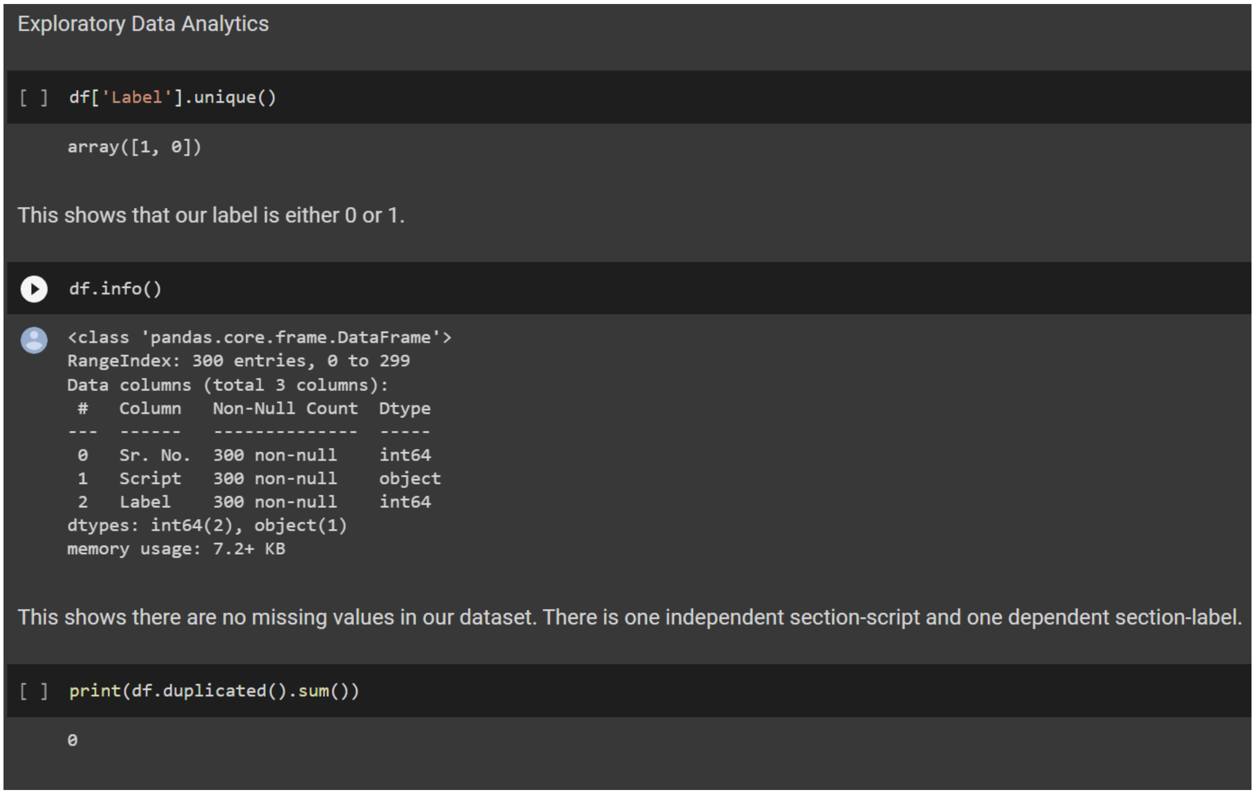
```
[ ] df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Indian Cyberbullying Datasets.csv")
```

Raw Exploration of Data

```
df
```

	Sr. No.	Script	Label
0	1	Kuch samay aur aage aajao, Sati bhi hota tha b...	1
1	2	The fact that the women in the past didn't use...	1
2	3	Madam plz read holy Quran and holy Bhagavad Gi...	0
3	4	So evolution doesn't mean anything? It was a c...	0
4	5	Wese to sab pehele bandar the...to wese hi ghu...	1
...	...	...	...
295	296	Bhai Nepal mae it's damn expensive rs 1000 :(	0
296	297	bohot khushnaseebi hamaari, sway your big fat ...	1
297	298	machines dekh ke maja aa gaya,, errr you wr al...	0
298	299	only in power have the rights,, human or inhuman	0
299	300	sir ji some times make time to address youths ...	0

300 rows × 3 columns



Exploratory Data Analytics

```
[ ] df['Label'].unique()
```

```
array([1, 0])
```

This shows that our label is either 0 or 1.

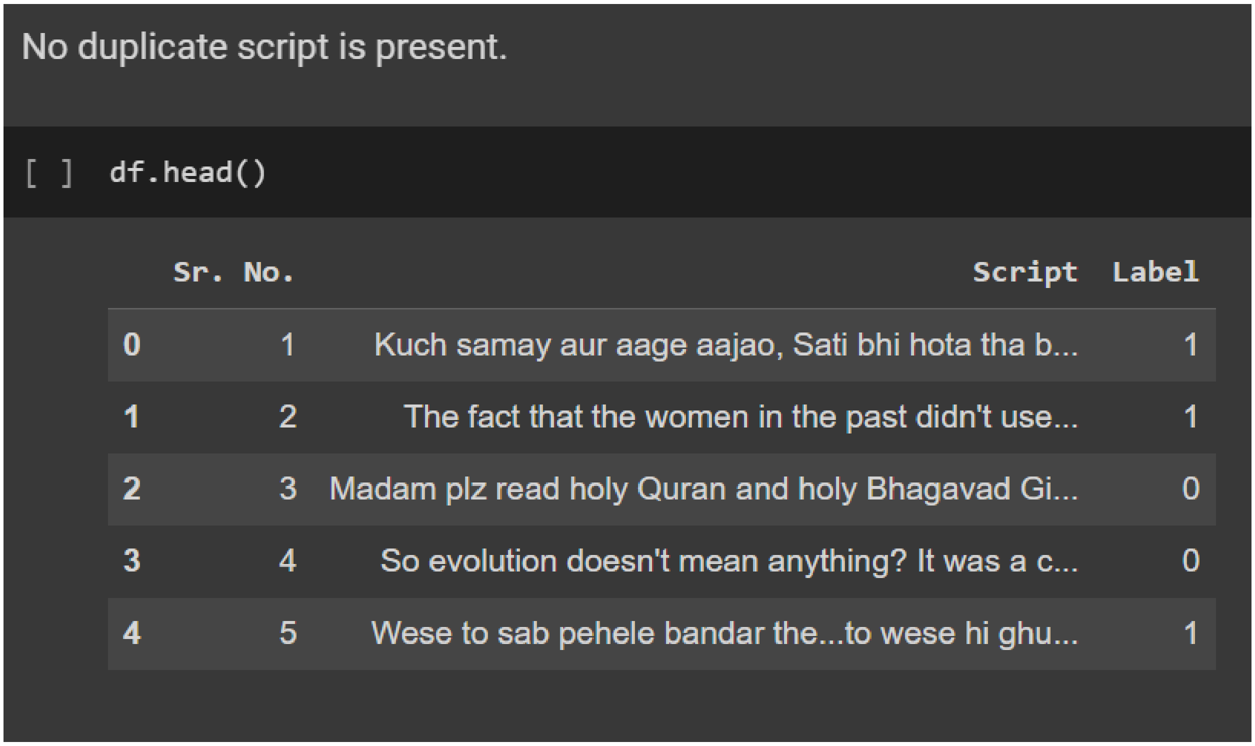
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Sr. No.   300 non-null    int64
 1   Script    300 non-null    object
 2   Label     300 non-null    int64
dtypes: int64(2), object(1)
memory usage: 7.2+ KB
```

This shows there are no missing values in our dataset. There is one independent section-script and one dependent section-label.

```
[ ] print(df.duplicated().sum())
```

```
0
```



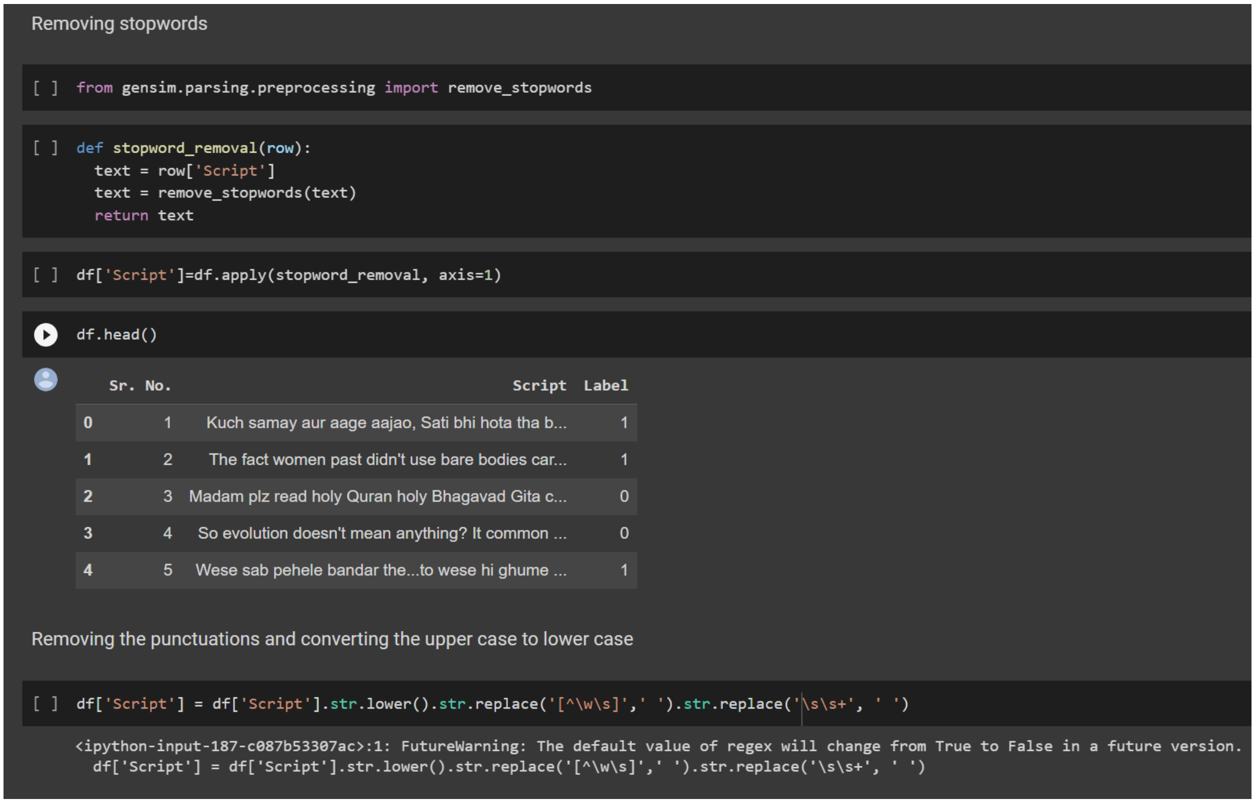
No duplicate script is present.

```
[ ] df.head()
```

	Sr. No.	Script	Label
0	1	Kuch samay aur aage aajao, Sati bhi hota tha b...	1
1	2	The fact that the women in the past didn't use...	1
2	3	Madam plz read holy Quran and holy Bhagavad Gi...	0
3	4	So evolution doesn't mean anything? It was a c...	0
4	5	Wese to sab pehele bandar the...to wese hi ghu...	1

II. Data Preprocessing

Stopwords-



Removing stopwords

```
[ ] from gensim.parsing.preprocessing import remove_stopwords
```

```
[ ] def stopword_removal(row):
      text = row['Script']
      text = remove_stopwords(text)
      return text
```

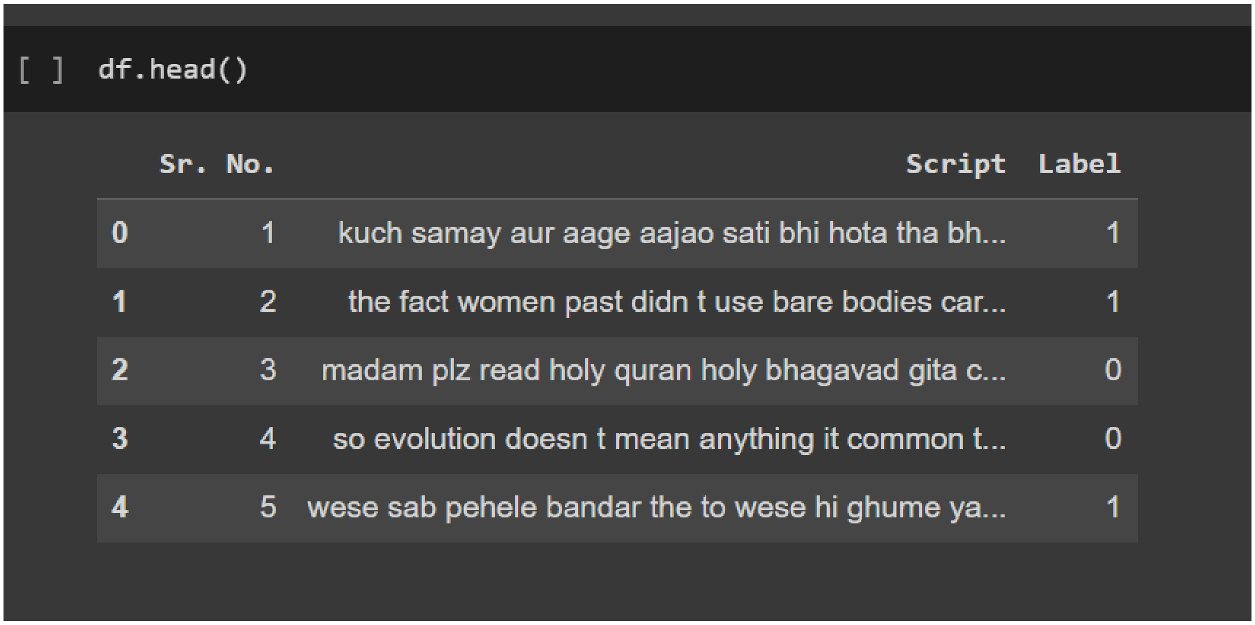
```
[ ] df['Script']=df.apply(stopword_removal, axis=1)
```

```
df.head()
```

	Sr. No.	Script	Label
0	1	Kuch samay aur aage aajao, Sati bhi hota tha b...	1
1	2	The fact women past didn't use bare bodies car...	1
2	3	Madam plz read holy Quran holy Bhagavad Gita c...	0
3	4	So evolution doesn't mean anything? It common ...	0
4	5	Wese sab pehele bandar the...to wese hi ghume ...	1

Removing the punctuations and converting the upper case to lower case

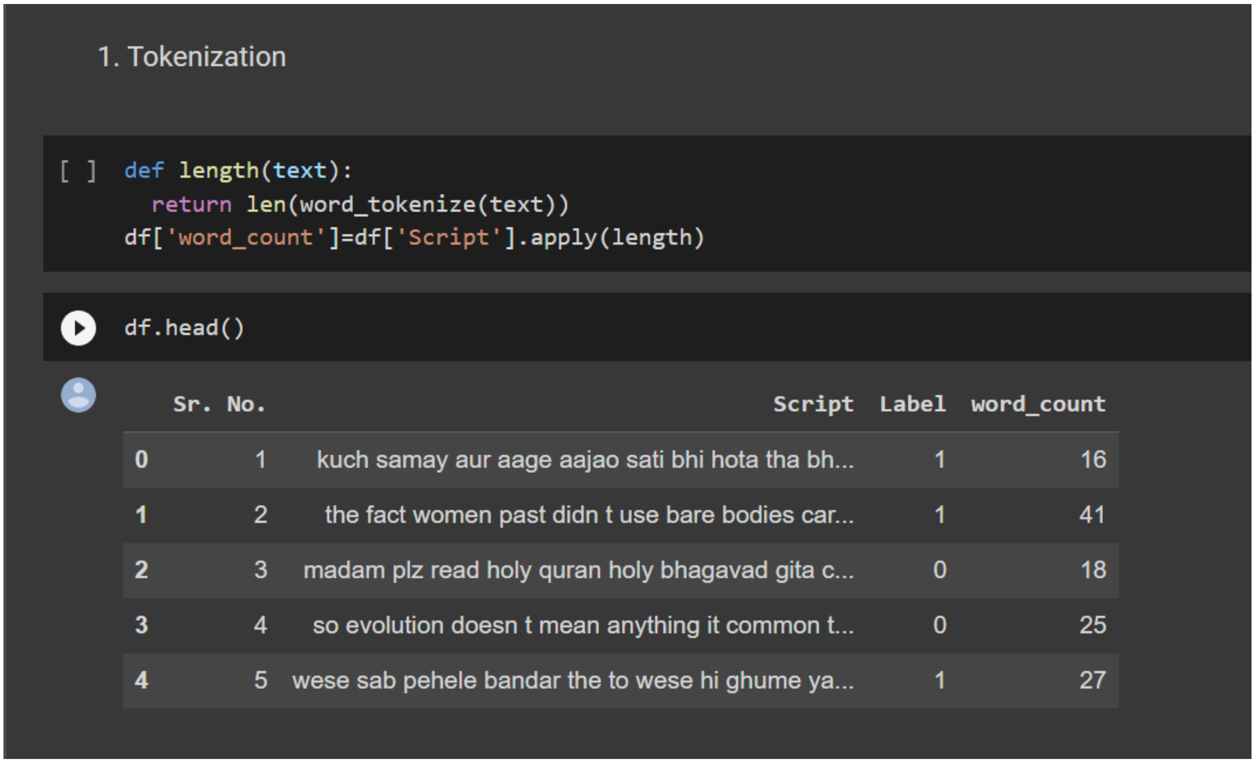
```
[ ] df['Script'] = df['Script'].str.lower().str.replace('[^\w\s]',' ').str.replace('\s\s+', ' ')

    <ipython-input-187-c087b53307ac>:1: FutureWarning: The default value of regex will change from True to False in a future version.
      df['Script'] = df['Script'].str.lower().str.replace('[^\w\s]',' ').str.replace('\s\s+', ' ')
```



```
[ ] df.head()
```

	Sr. No.	Script	Label
0	1	kuch samay aur aage aajao sati bhi hota tha bh...	1
1	2	the fact women past didn t use bare bodies car...	1
2	3	madam plz read holy quran holy bhagavad gita c...	0
3	4	so evolution doesn t mean anything it common t...	0
4	5	wese sab pehele bandar the to wese hi ghume ya...	1

Tokenization-



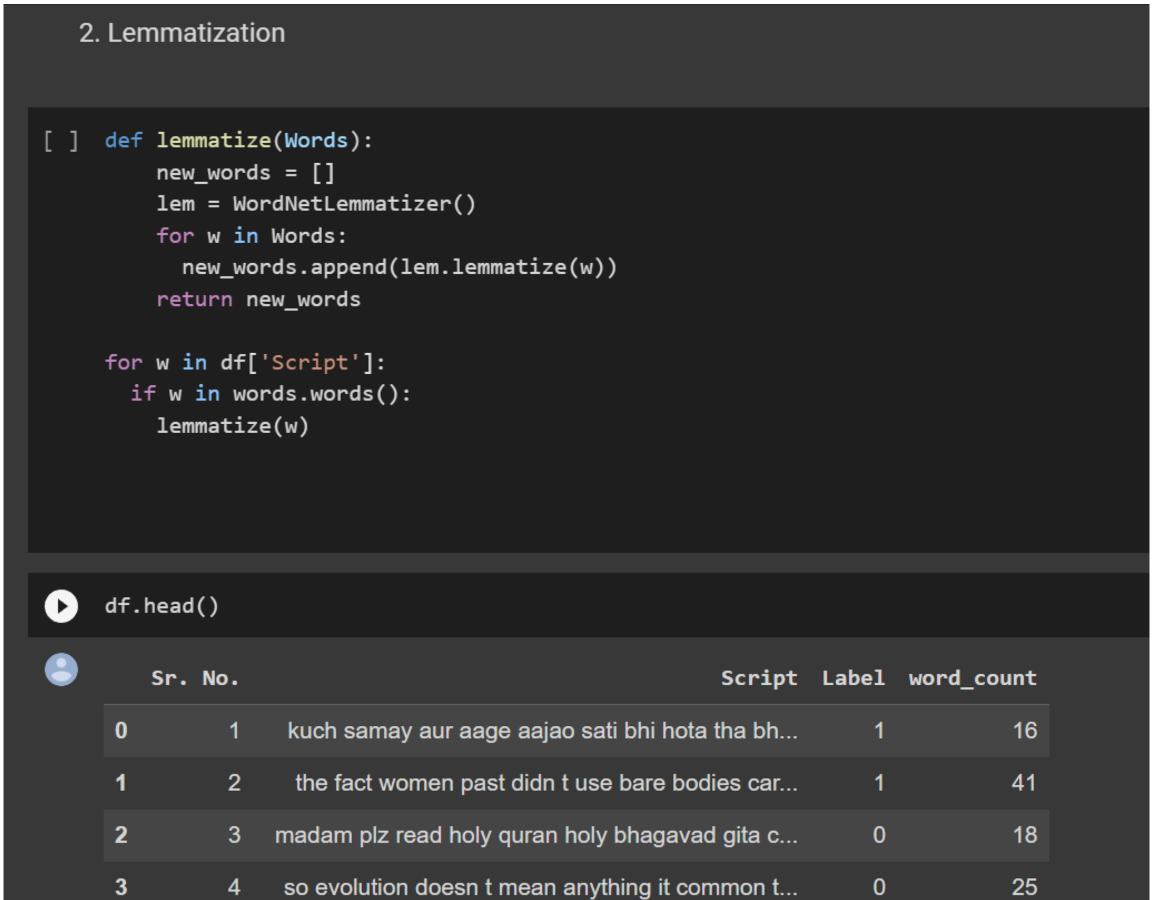
1. Tokenization

```
def length(text):
  return len(word_tokenize(text))
df['word_count']=df['Script'].apply(length)
```

```
df.head()
```

	Sr. No.	Script	Label	word_count
0	1	kuch samay aur aage aajao sati bhi hota tha bh...	1	16
1	2	the fact women past didn t use bare bodies car...	1	41
2	3	madam plz read holy quran holy bhagavad gita c...	0	18
3	4	so evolution doesn t mean anything it common t...	0	25
4	5	wese sab pehele bandar the to wese hi ghume ya...	1	27

Lemmatization-



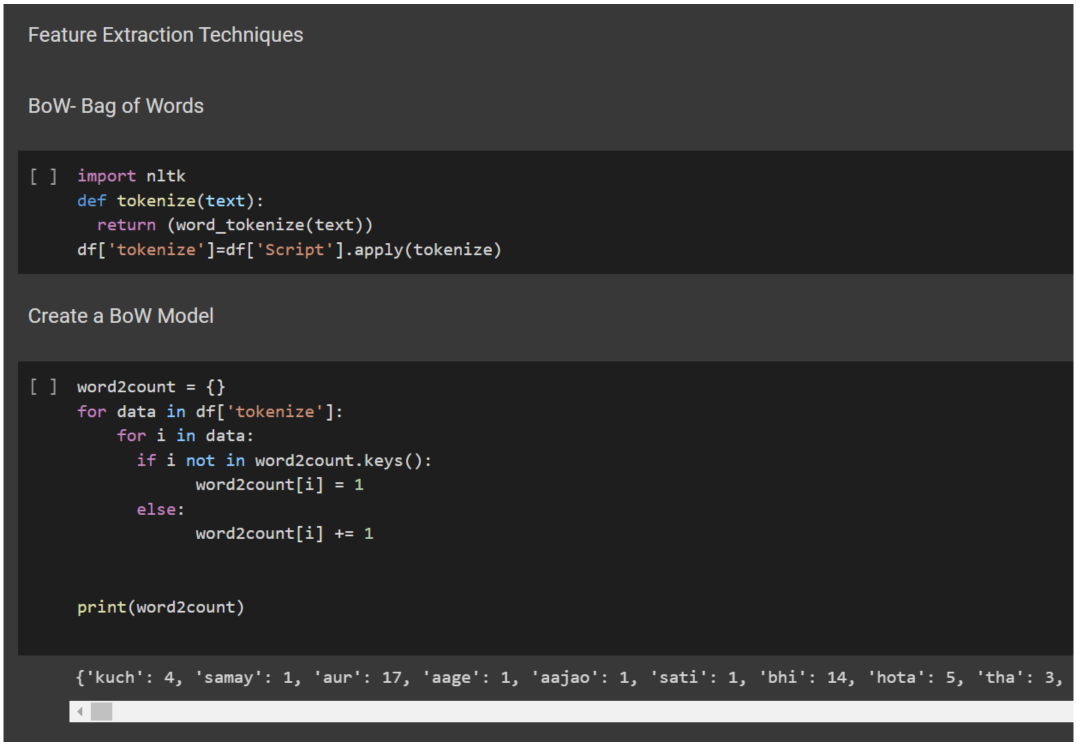
2. Lemmatization

```
def lemmatize(Words):
    new_words = []
    lem = WordNetLemmatizer()
    for w in Words:
      new_words.append(lem.lemmatize(w))
    return new_words

for w in df['Script']:
  if w in words.words():
    lemmatize(w)
```

```
df.head()
```

	Sr. No.	Script	Label	word_count
0	1	kuch samay aur aage aajao sati bhi hota tha bh...	1	16
1	2	the fact women past didn t use bare bodies car...	1	41
2	3	madam plz read holy quran holy bhagavad gita c...	0	18
3	4	so evolution doesn t mean anything it common t...	0	25

III. Feature Extraction Techniques



Feature Extraction Techniques

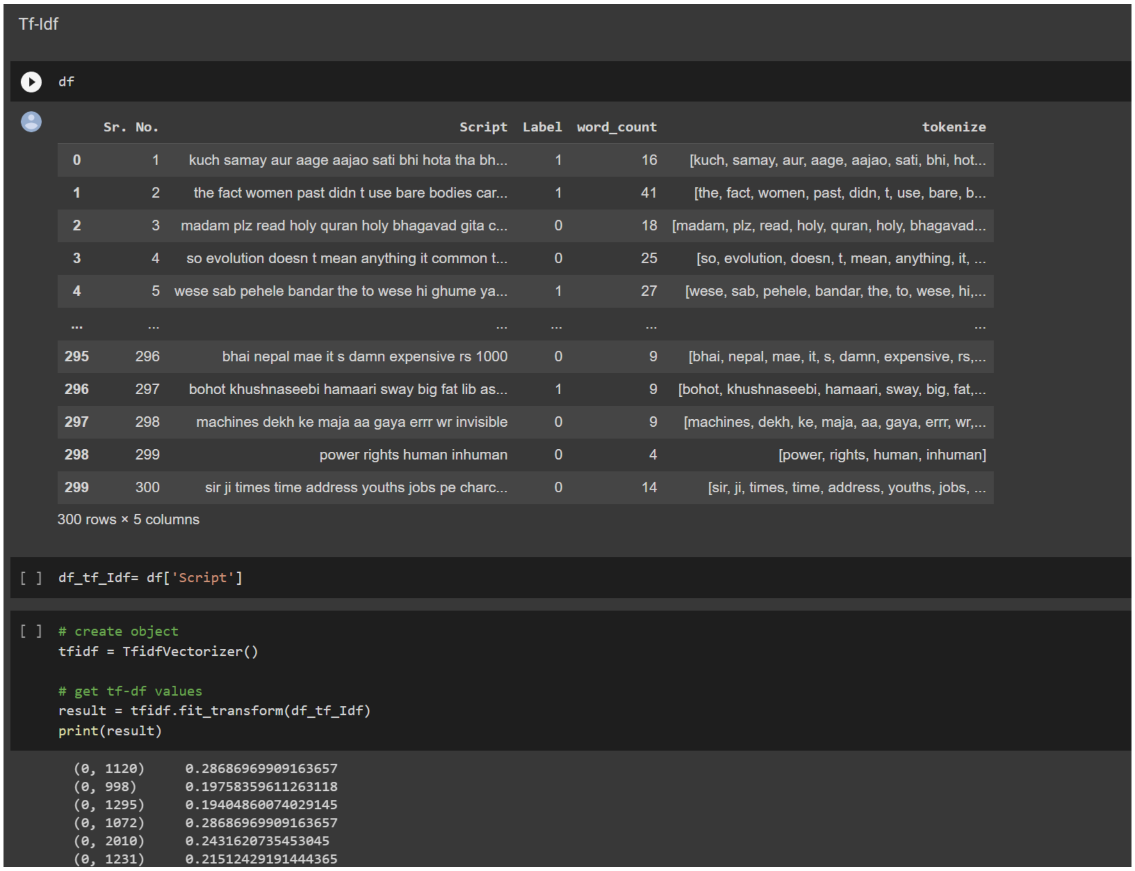
BoW- Bag of Words

```
[ ]  import nltk
     def tokenize(text):
       return (word_tokenize(text))
     df['tokenize']=df['Script'].apply(tokenize)
```

Create a BoW Model

```
[ ]  word2count = {}
     for data in df['tokenize']:
         for i in data:
           if i not in word2count.keys():
                 word2count[i] = 1
           else:
                 word2count[i] += 1


     print(word2count)
```

```
{'kuch': 4, 'samay': 1, 'aur': 17, 'aage': 1, 'aajao': 1, 'sati': 1, 'bhi': 14, 'hota': 5, 'tha': 3,
```



Tf-Idf

```
df
```

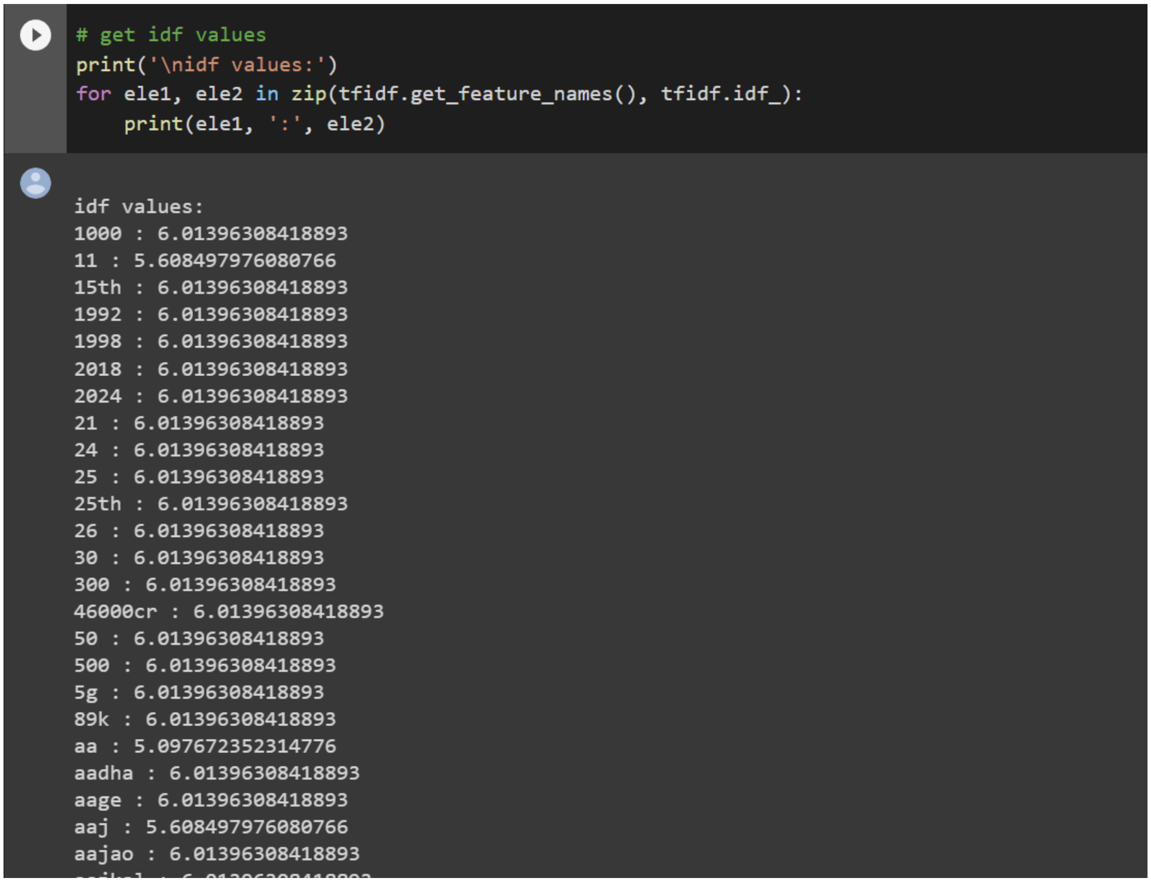
	Sr. No.	Script	Label	word_count	tokenize
0	1	kuch samay aur aage aajao sati bhi hota tha bh...	1	16	[kuch, samay, aur, aage, aajao, sati, bhi, hot...
1	2	the fact women past didn t use bare bodies car...	1	41	[the, fact, women, past, didn, t, use, bare, b...
2	3	madam plz read holy quran holy bhagavad gita c...	0	18	[madam, plz, read, holy, quran, holy, bhagavad...
3	4	so evolution doesn t mean anything it common t...	0	25	[so, evolution, doesn, t, mean, anything, it, ...
4	5	wese sab pehele bandar the to wese hi ghume ya...	1	27	[wese, sab, pehele, bandar, the, to, wese, hi,...
...	...	...	...	...	...
295	296	bhai nepal mae it s damn expensive rs 1000	0	9	[bhai, nepal, mae, it, s, damn, expensive, rs,...
296	297	bohot khushnaseebi hamaari sway big fat lib as...	1	9	[bohot, khushnaseebi, hamaari, sway, big, fat,...
297	298	machines dekh ke maja aa gaya errr wr invisible	0	9	[machines, dekh, ke, maja, aa, gaya, errr, wr,...
298	299	power rights human inhuman	0	4	[power, rights, human, inhuman]
299	300	sir ji times time address youths jobs pe charc...	0	14	[sir, ji, times, time, address, youths, jobs, ...

300 rows × 5 columns

```
[ ]  df_tf_Idf= df['Script']
```

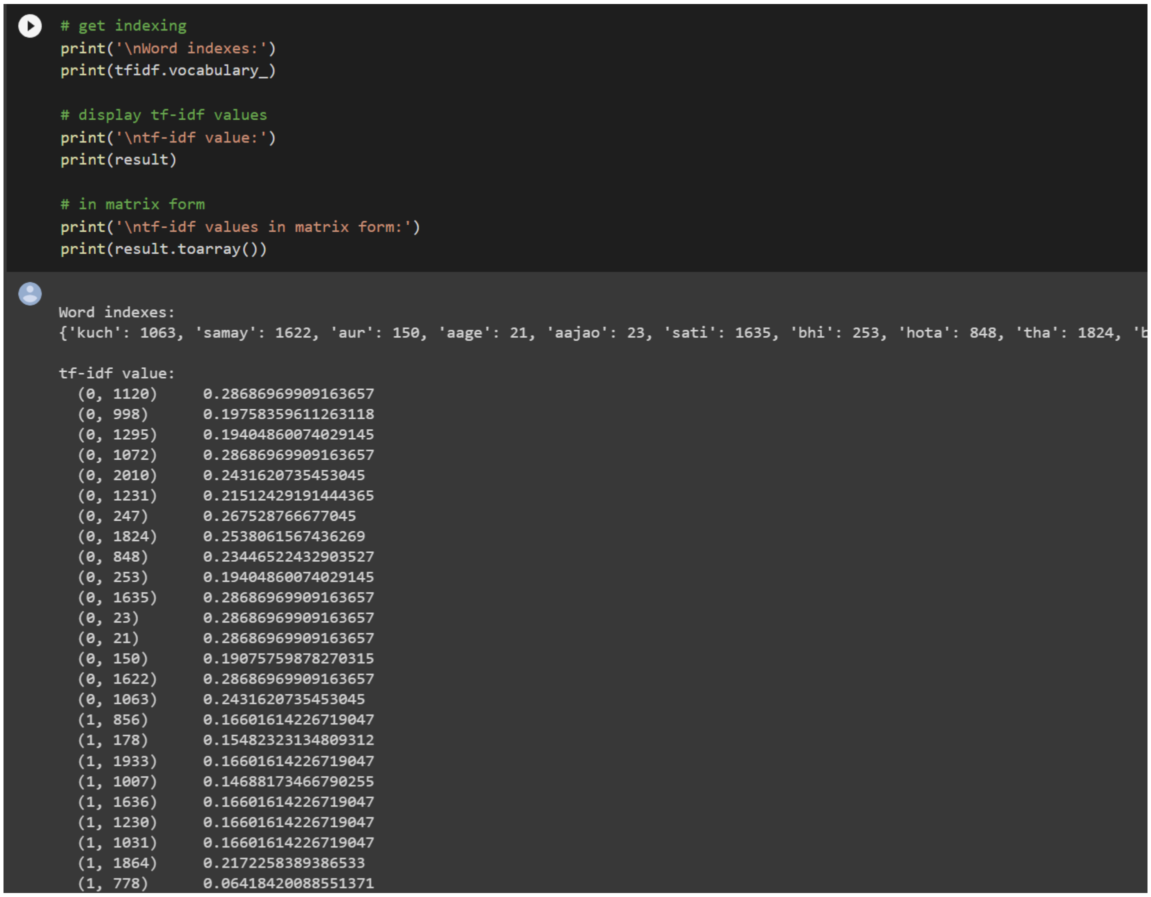
```
[ ]  # create object
     tfidf = TfidfVectorizer()

     # get tf-df values
     result = tfidf.fit_transform(df_tf_Idf)
     print(result)
```

```
  (0, 1120)	0.28686969909163657
  (0, 998)	0.19758359611263118
  (0, 1295)	0.19404860074029145
  (0, 1072)	0.28686969909163657
  (0, 2010)	0.2431620735453045
  (0, 1231)	0.21512429191444365
```



```python
# get idf values
print('\nidf values:')
for ele1, ele2 in zip(tfidf.get_feature_names(), tfidf.idf_):
    print(ele1, ':', ele2)
```

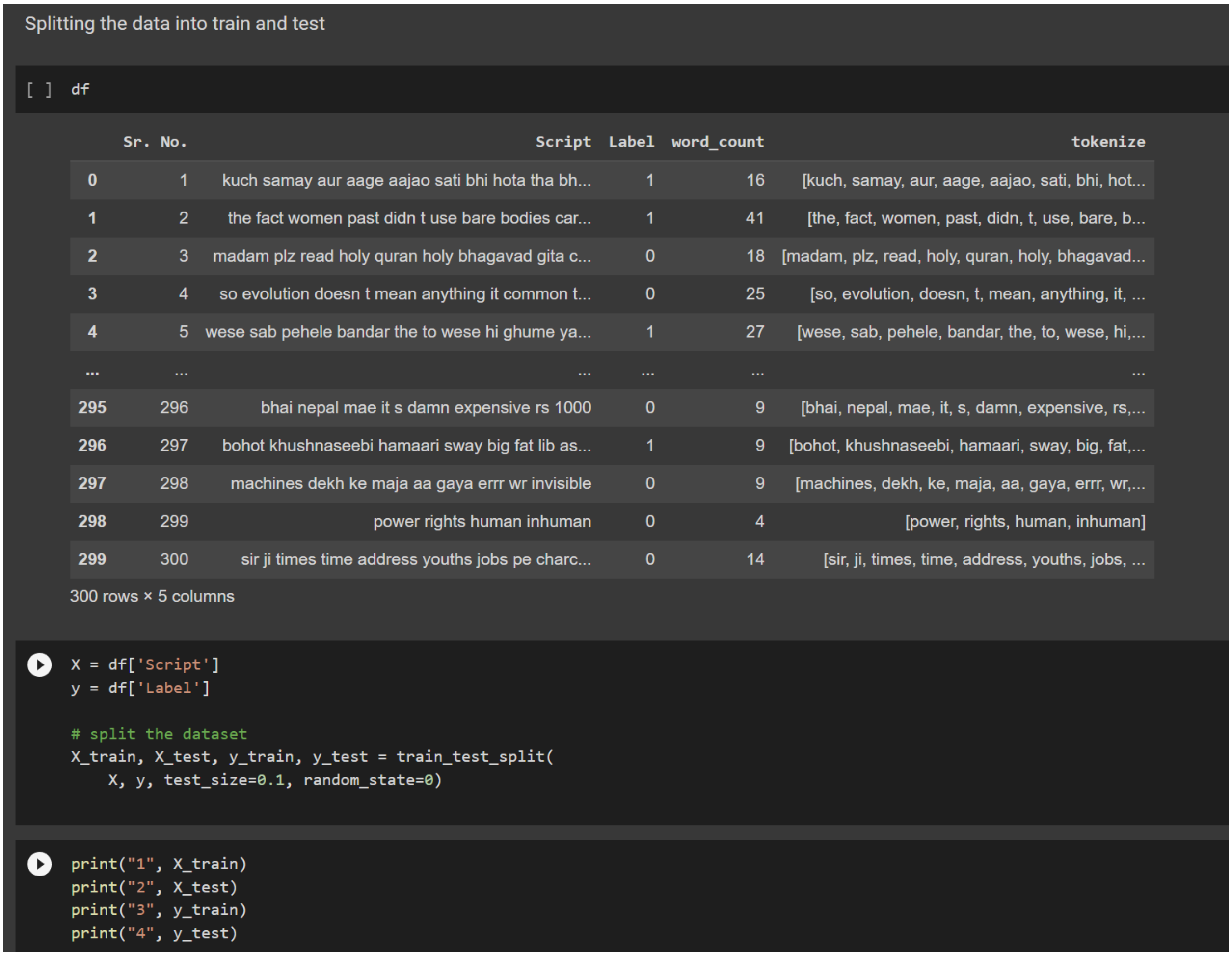
```
idf values:
1000 : 6.01396308418893
11 : 5.608497976080766
15th : 6.01396308418893
1992 : 6.01396308418893
1998 : 6.01396308418893
2018 : 6.01396308418893
2024 : 6.01396308418893
21 : 6.01396308418893
24 : 6.01396308418893
25 : 6.01396308418893
25th : 6.01396308418893
26 : 6.01396308418893
30 : 6.01396308418893
300 : 6.01396308418893
46000cr : 6.01396308418893
50 : 6.01396308418893
500 : 6.01396308418893
5g : 6.01396308418893
89k : 6.01396308418893
aa : 5.097672352314776
aadha : 6.01396308418893
aage : 6.01396308418893
aaj : 5.608497976080766
aajao : 6.01396308418893
```



```python
# get indexing
print('\nWord indexes:')
print(tfidf.vocabulary_)

# display tf-idf values
print('\ntf-idf value:')
print(result)

# in matrix form
print('\ntf-idf values in matrix form:')
print(result.toarray())
```

```
Word indexes:
{'kuch': 1063, 'samay': 1622, 'aur': 150, 'aage': 21, 'aajao': 23, 'sati': 1635, 'bhi': 253, 'hota': 848, 'tha': 1824, 'b

tf-idf value:
  (0, 1120)	0.28686969909163657
  (0, 998)	0.19758359611263118
  (0, 1295)	0.19404860074029145
  (0, 1072)	0.28686969909163657
  (0, 2010)	0.2431620735453045
  (0, 1231)	0.21512429191444365
  (0, 247)	0.267528766677045
  (0, 1824)	0.2538061567436269
  (0, 848)	0.23446522432903527
  (0, 253)	0.19404860074029145
  (0, 1635)	0.28686969909163657
  (0, 23)	0.28686969909163657
  (0, 21)	0.28686969909163657
  (0, 150)	0.19075759878270315
  (0, 1622)	0.28686969909163657
  (0, 1063)	0.2431620735453045
  (1, 856)	0.16601614226719047
  (1, 178)	0.15482323134809312
  (1, 1933)	0.16601614226719047
  (1, 1007)	0.14688173466790255
  (1, 1636)	0.16601614226719047
  (1, 1230)	0.16601614226719047
  (1, 1031)	0.16601614226719047
  (1, 1864)	0.2172258389386533
  (1, 778)	0.06418420088551371
```

IV. Splitting the data



Splitting the data into train and test

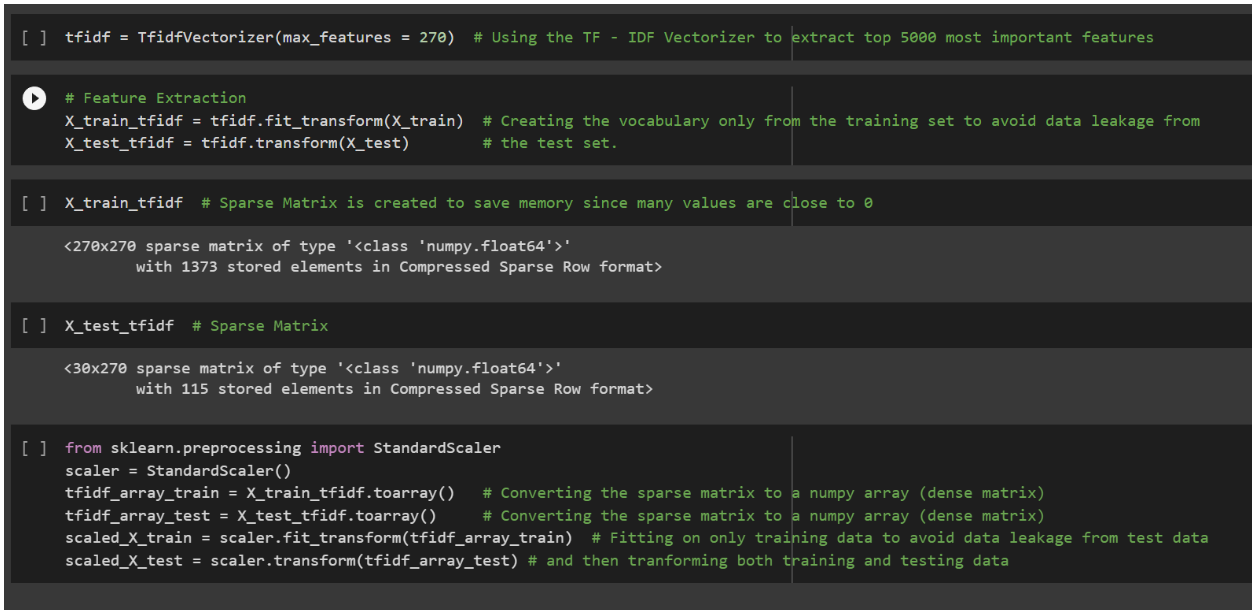
```
[ ] df
```

	Sr. No.	Script	Label	word_count	tokenize
0	1	kuch samay aur aage aajao sati bhi hota tha bh...	1	16	[kuch, samay, aur, aage, aajao, sati, bhi, hot...
1	2	the fact women past didn t use bare bodies car...	1	41	[the, fact, women, past, didn, t, use, bare, b...
2	3	madam plz read holy quran holy bhagavad gita c...	0	18	[madam, plz, read, holy, quran, holy, bhagavad...
3	4	so evolution doesn t mean anything it common t...	0	25	[so, evolution, doesn, t, mean, anything, it, ...
4	5	wese sab pehele bandar the to wese hi ghume ya...	1	27	[wese, sab, pehele, bandar, the, to, wese, hi,...
...	...	...	...	...	...
295	296	bhai nepal mae it s damn expensive rs 1000	0	9	[bhai, nepal, mae, it, s, damn, expensive, rs,...
296	297	bohot khushnaseebi hamaari sway big fat lib as...	1	9	[bohot, khushnaseebi, hamaari, sway, big, fat,...
297	298	machines dekh ke maja aa gaya errr wr invisible	0	9	[machines, dekh, ke, maja, aa, gaya, errr, wr,...
298	299	power rights human inhuman	0	4	[power, rights, human, inhuman]
299	300	sir ji times time address youths jobs pe charc...	0	14	[sir, ji, times, time, address, youths, jobs, ...

300 rows × 5 columns

```
X = df['Script']
y = df['Label']

# split the dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.1, random_state=0)
```

```
print("1", X_train)
print("2", X_test)
print("3", y_train)
print("4", y_test)
```



```
[ ] tfidf = TfidfVectorizer(max_features = 270)  # Using the TF - IDF Vectorizer to extract top 5000 most important features
```

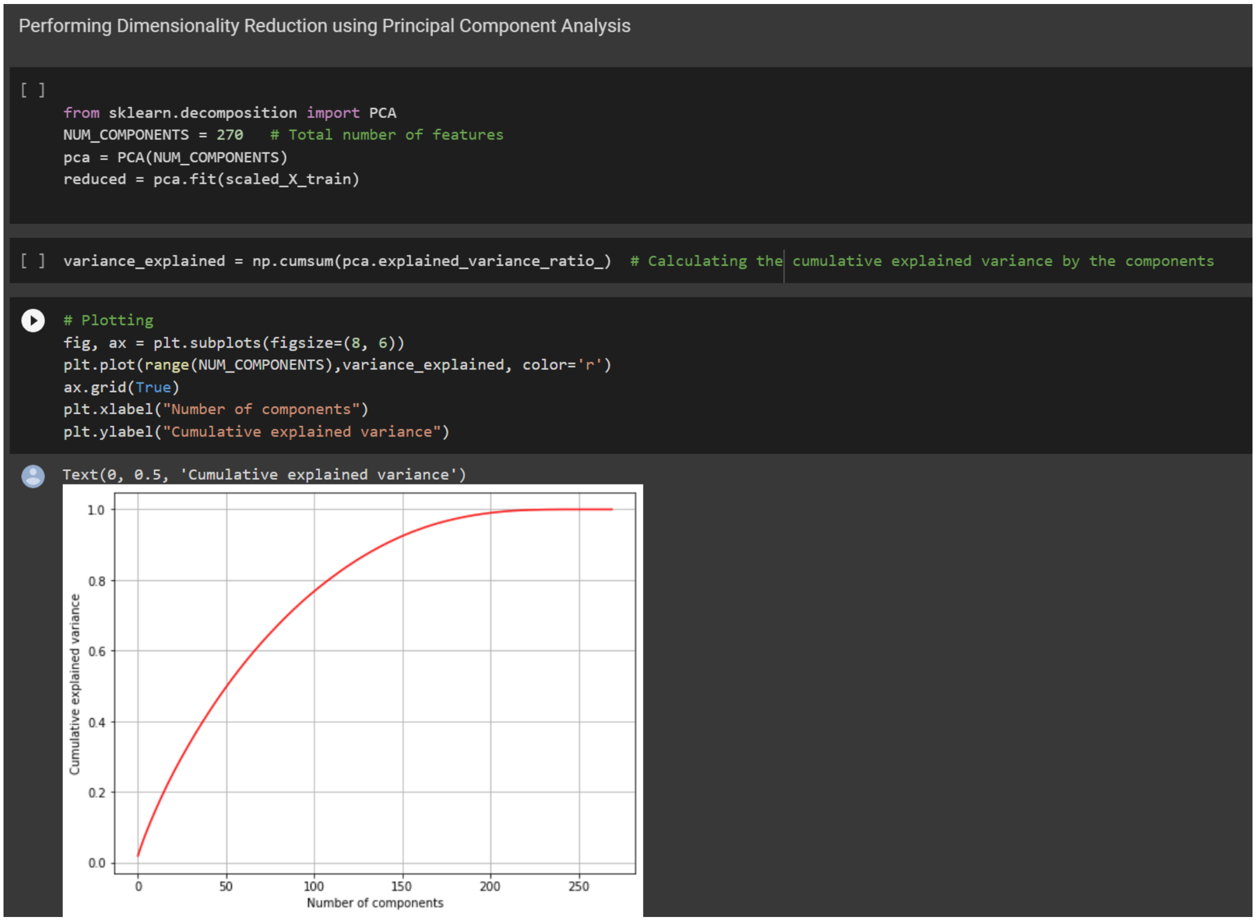
```
# Feature Extraction
X_train_tfidf = tfidf.fit_transform(X_train)  # Creating the vocabulary only from the training set to avoid data leakage from
X_test_tfidf = tfidf.transform(X_test)        # the test set.
```

```
[ ] X_train_tfidf  # Sparse Matrix is created to save memory since many values are close to 0

<270x270 sparse matrix of type '<class 'numpy.float64'>'
        with 1373 stored elements in Compressed Sparse Row format>
```

```
[ ] X_test_tfidf  # Sparse Matrix

<30x270 sparse matrix of type '<class 'numpy.float64'>'
        with 115 stored elements in Compressed Sparse Row format>
```

```
[ ] from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
tfidf_array_train = X_train_tfidf.toarray()   # Converting the sparse matrix to a numpy array (dense matrix)
tfidf_array_test = X_test_tfidf.toarray()     # Converting the sparse matrix to a numpy array (dense matrix)
scaled_X_train = scaler.fit_transform(tfidf_array_train)  # Fitting on only training data to avoid data leakage from test data
scaled_X_test = scaler.transform(tfidf_array_test) # and then tranforming both training and testing data
```

V. Principal Component Analysis



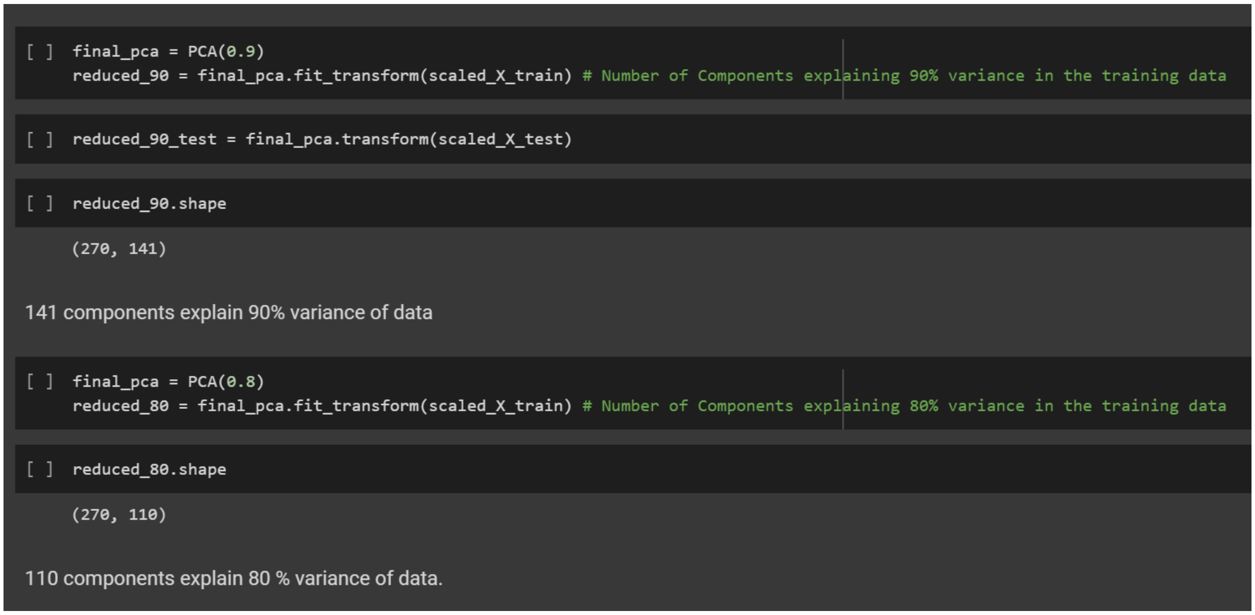
Performing Dimensionality Reduction using Principal Component Analysis

```
[ ]
    from sklearn.decomposition import PCA
    NUM_COMPONENTS = 270   # Total number of features
    pca = PCA(NUM_COMPONENTS)
    reduced = pca.fit(scaled_X_train)
```

```
[ ] variance_explained = np.cumsum(pca.explained_variance_ratio_)  # Calculating the cumulative explained variance by the components
```

```
    # Plotting
    fig, ax = plt.subplots(figsize=(8, 6))
    plt.plot(range(NUM_COMPONENTS),variance_explained, color='r')
    ax.grid(True)
    plt.xlabel("Number of components")
    plt.ylabel("Cumulative explained variance")
```

```
Text(0, 0.5, 'Cumulative explained variance')
```



```
[ ] final_pca = PCA(0.9)
    reduced_90 = final_pca.fit_transform(scaled_X_train) # Number of Components explaining 90% variance in the training data
```

```
[ ] reduced_90_test = final_pca.transform(scaled_X_test)
```

```
[ ] reduced_90.shape

    (270, 141)
```

141 components explain 90% variance of data

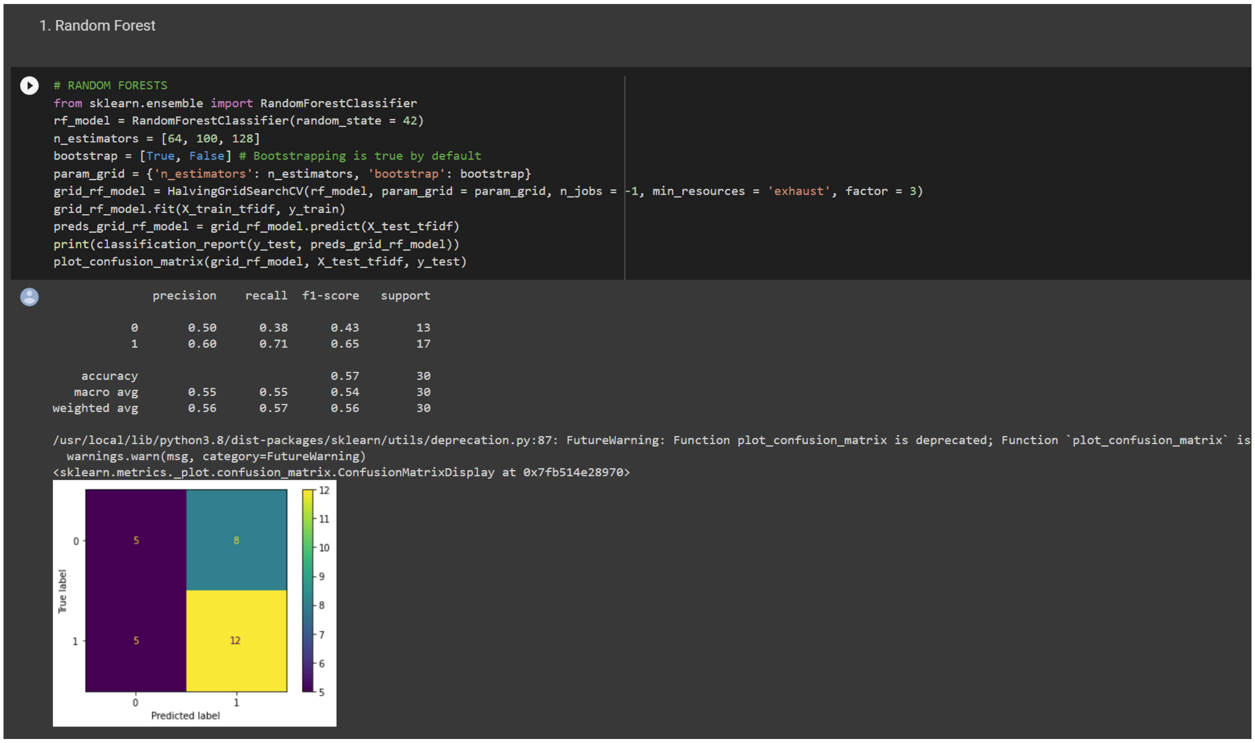
```
[ ] final_pca = PCA(0.8)
    reduced_80 = final_pca.fit_transform(scaled_X_train) # Number of Components explaining 80% variance in the training data
```

```
[ ] reduced_80.shape

    (270, 110)
```

110 components explain 80 % variance of data.

VI. Model Training

1. Random Forest

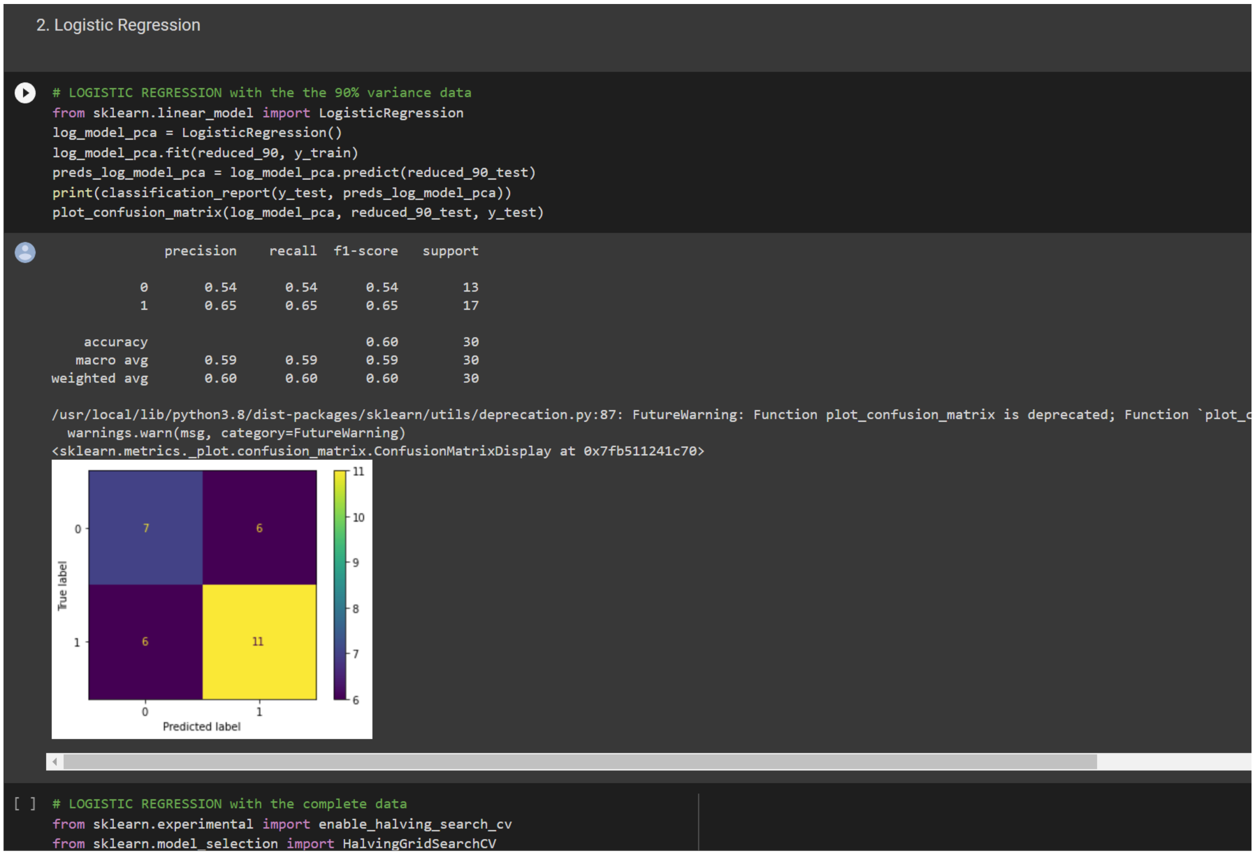


1. Random Forest

```
# RANDOM FORESTS
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(random_state = 42)
n_estimators = [64, 100, 128]
bootstrap = [True, False] # Bootstrapping is true by default
param_grid = {'n_estimators': n_estimators, 'bootstrap': bootstrap}
grid_rf_model = HalvingGridSearchCV(rf_model, param_grid = param_grid, n_jobs = -1, min_resources = 'exhaust', factor = 3)
grid_rf_model.fit(X_train_tfidf, y_train)
preds_grid_rf_model = grid_rf_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_grid_rf_model))
plot_confusion_matrix(grid_rf_model, X_test_tfidf, y_test)
```

```
              precision    recall  f1-score   support

           0       0.50      0.38      0.43        13
           1       0.60      0.71      0.65        17

    accuracy                           0.57        30
   macro avg       0.55      0.55      0.54        30
weighted avg       0.56      0.57      0.56        30

/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fb514e28970>
```

2. Logistic Regression

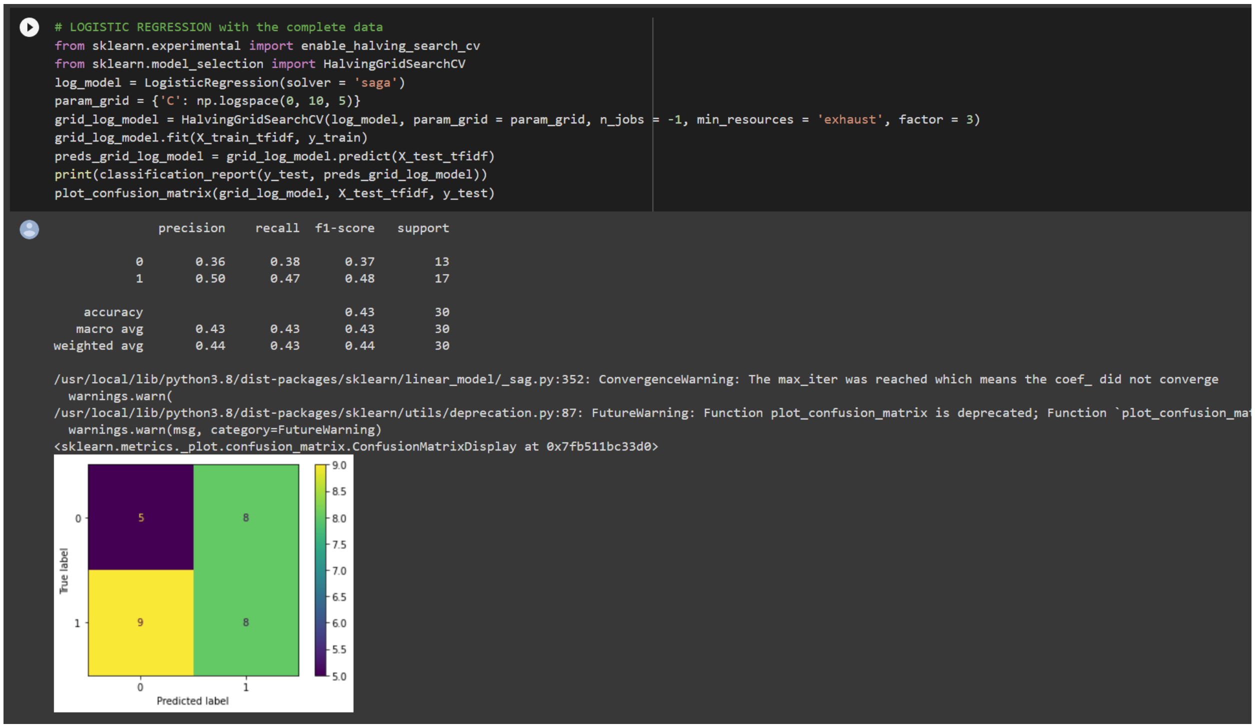


2. Logistic Regression

```
# LOGISTIC REGRESSION with the the 90% variance data
from sklearn.linear_model import LogisticRegression
log_model_pca = LogisticRegression()
log_model_pca.fit(reduced_90, y_train)
preds_log_model_pca = log_model_pca.predict(reduced_90_test)
print(classification_report(y_test, preds_log_model_pca))
plot_confusion_matrix(log_model_pca, reduced_90_test, y_test)
```

```
              precision    recall  f1-score   support

           0       0.54      0.54      0.54        13
           1       0.65      0.65      0.65        17

    accuracy                           0.60        30
   macro avg       0.59      0.59      0.59        30
weighted avg       0.60      0.60      0.60        30

/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fb511241c70>
```

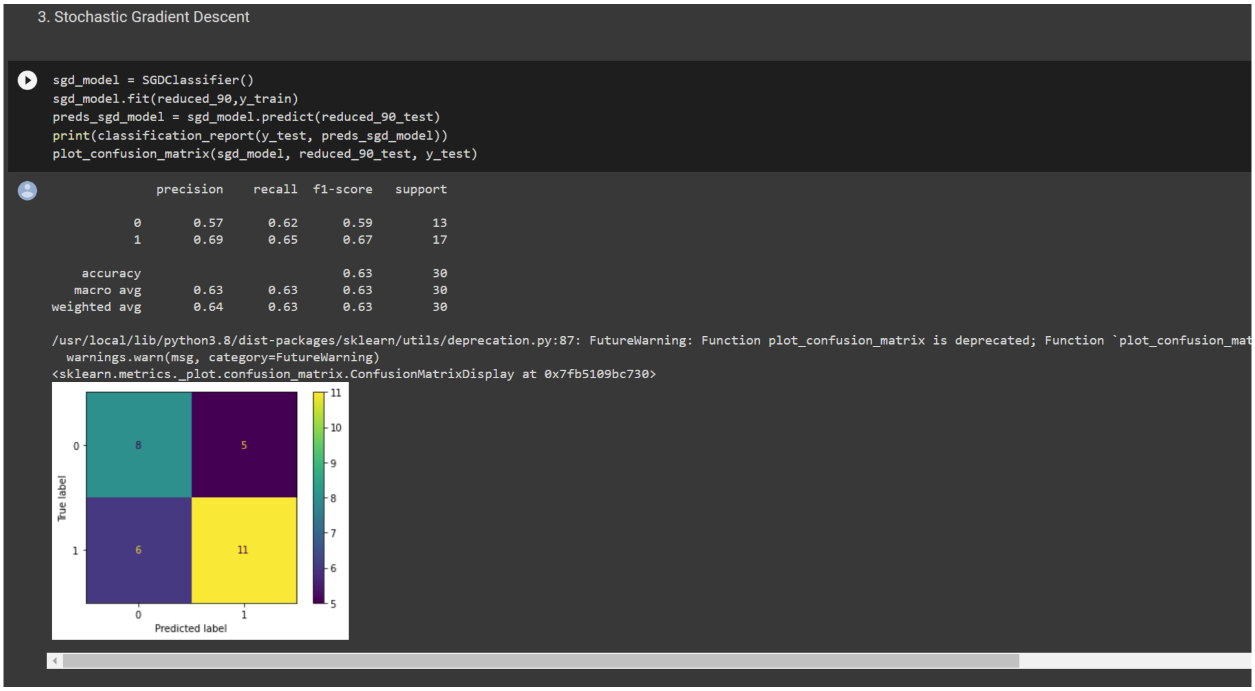
```
[ ] # LOGISTIC REGRESSION with the complete data
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
```



```
# LOGISTIC REGRESSION with the complete data
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
log_model = LogisticRegression(solver = 'saga')
param_grid = {'C': np.logspace(0, 10, 5)}
grid_log_model = HalvingGridSearchCV(log_model, param_grid = param_grid, n_jobs = -1, min_resources = 'exhaust', factor = 3)
grid_log_model.fit(X_train_tfidf, y_train)
preds_grid_log_model = grid_log_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_grid_log_model))
plot_confusion_matrix(grid_log_model, X_test_tfidf, y_test)
```

```
              precision    recall  f1-score   support

           0       0.36      0.38      0.37        13
           1       0.50      0.47      0.48        17

    accuracy                           0.43        30
   macro avg       0.43      0.43      0.43        30
weighted avg       0.44      0.43      0.44        30

/usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_sag.py:352: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_mat
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fb511bc33d0>
```

3. Stochastic Gradient Search

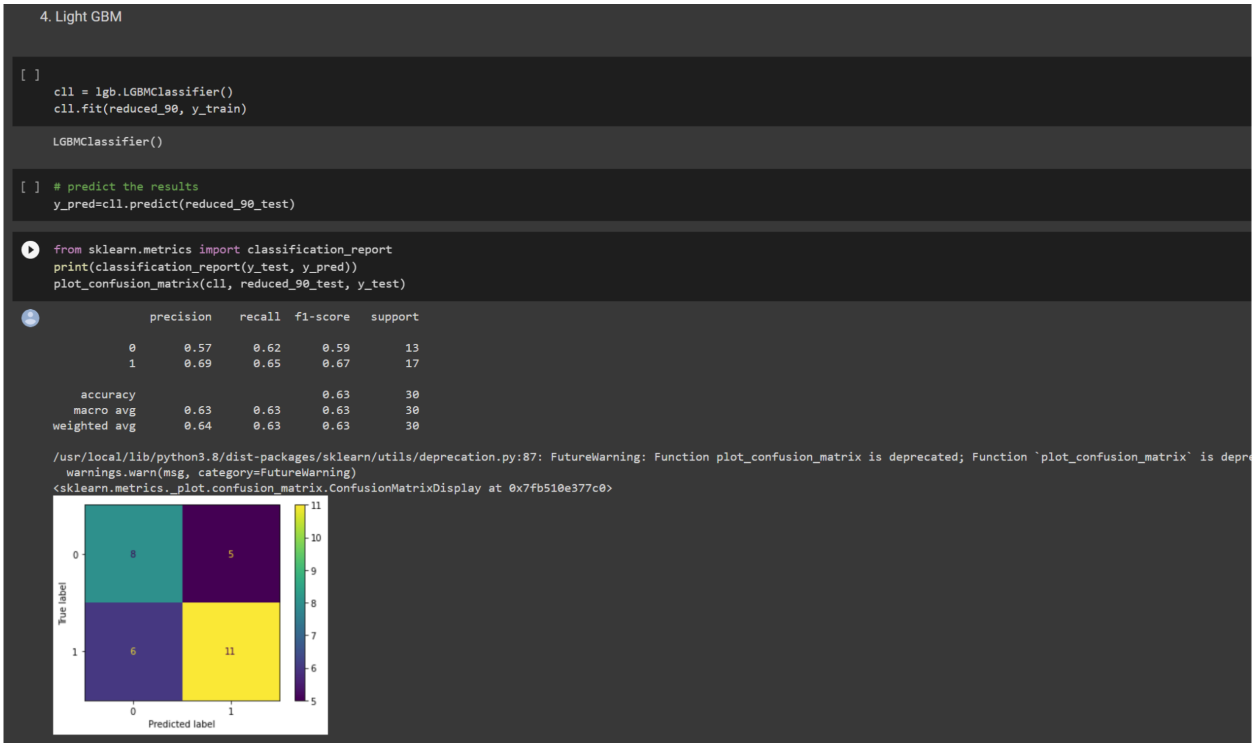


3. Stochastic Gradient Descent

```
sgd_model = SGDClassifier()
sgd_model.fit(reduced_90,y_train)
preds_sgd_model = sgd_model.predict(reduced_90_test)
print(classification_report(y_test, preds_sgd_model))
plot_confusion_matrix(sgd_model, reduced_90_test, y_test)
```

```
              precision    recall  f1-score   support

           0       0.57      0.62      0.59        13
           1       0.69      0.65      0.67        17

    accuracy                           0.63        30
   macro avg       0.63      0.63      0.63        30
weighted avg       0.64      0.63      0.63        30

/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_mat
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fb5109bc730>
```

4. Light GBM



4. Light GBM

```
[ ]
cl1 = lgb.LGBMClassifier()
cl1.fit(reduced_90, y_train)
```

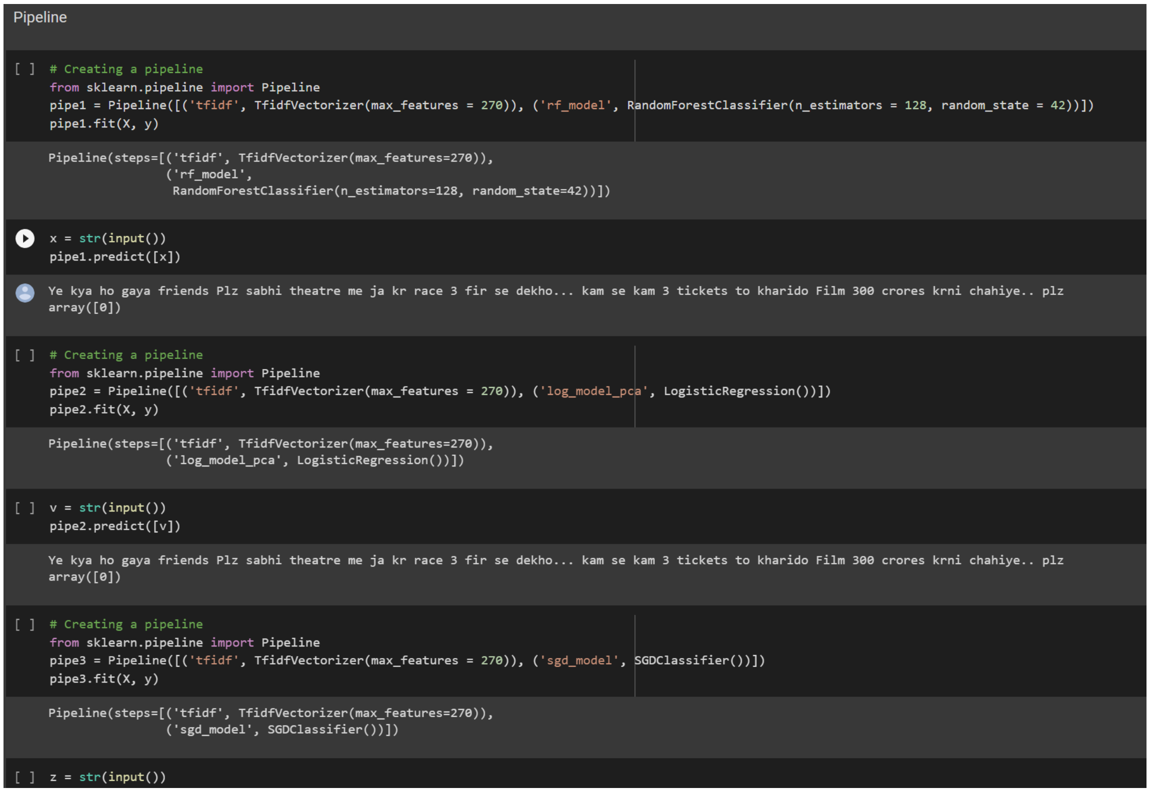
```
LGBMClassifier()
```

```
[ ] # predict the results
y_pred=cl1.predict(reduced_90_test)
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
plot_confusion_matrix(cl1, reduced_90_test, y_test)
```

```
              precision    recall  f1-score   support

           0       0.57      0.62      0.59        13
           1       0.69      0.65      0.67        17

    accuracy                           0.63        30
   macro avg       0.63      0.63      0.63        30
weighted avg       0.64      0.63      0.63        30

/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is depre
  warnings.warn(msg, category=FutureWarning)
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fb510e377c0>
```

VII. Pipeline



Pipeline

```
# Creating a pipeline
from sklearn.pipeline import Pipeline
pipe1 = Pipeline([('tfidf', TfidfVectorizer(max_features = 270)), ('rf_model', RandomForestClassifier(n_estimators = 128, random_state = 42))])
pipe1.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer(max_features=270)),
                ('rf_model',
                 RandomForestClassifier(n_estimators=128, random_state=42))])
```

```
x = str(input())
pipe1.predict([x])
```

```
Ye kya ho gaya friends Plz sabhi theatre me ja kr race 3 fir se dekho... kam se kam 3 tickets to kharido Film 300 crores krni chahiye.. plz
array([0])
```

```
# Creating a pipeline
from sklearn.pipeline import Pipeline
pipe2 = Pipeline([('tfidf', TfidfVectorizer(max_features = 270)), ('log_model_pca', LogisticRegression())])
pipe2.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer(max_features=270)),
                ('log_model_pca', LogisticRegression())])
```

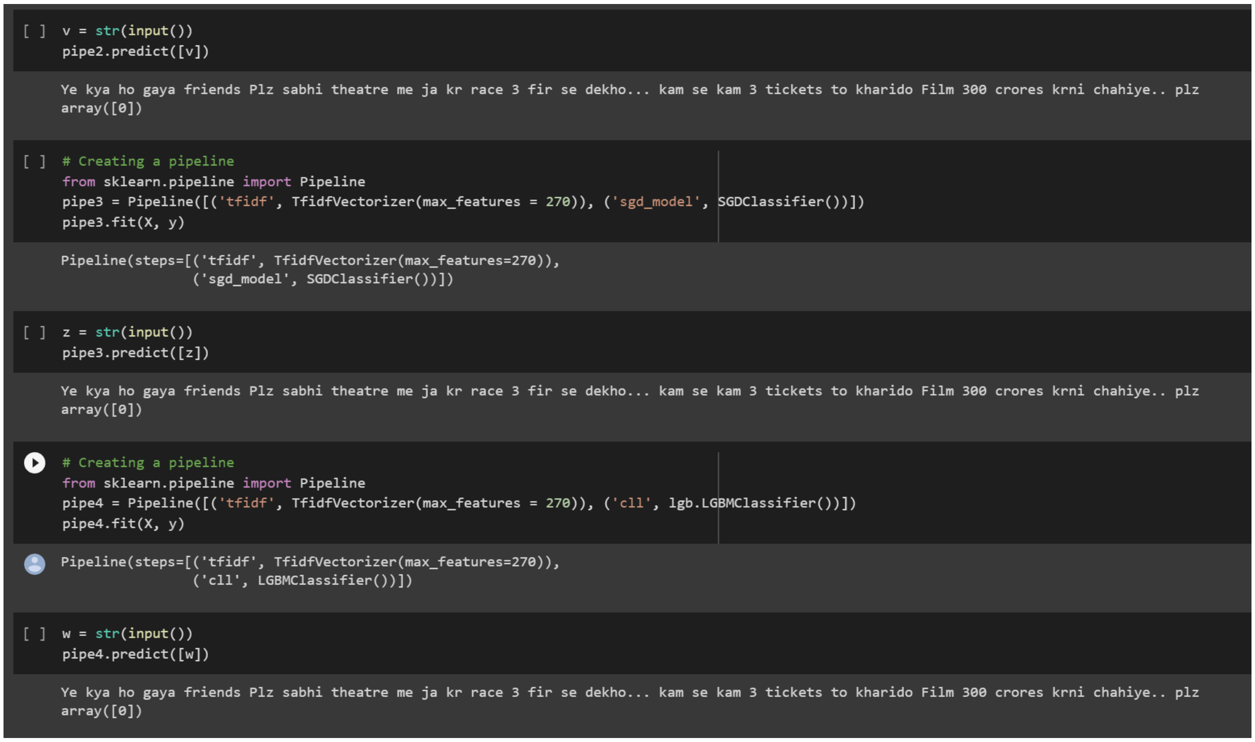
```
v = str(input())
pipe2.predict([v])
```

```
Ye kya ho gaya friends Plz sabhi theatre me ja kr race 3 fir se dekho... kam se kam 3 tickets to kharido Film 300 crores krni chahiye.. plz
array([0])
```

```
# Creating a pipeline
from sklearn.pipeline import Pipeline
pipe3 = Pipeline([('tfidf', TfidfVectorizer(max_features = 270)), ('sgd_model', SGDClassifier())])
pipe3.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer(max_features=270)),
                ('sgd_model', SGDClassifier())])
```

```
z = str(input())
```



```
v = str(input())
pipe2.predict([v])
```

```
Ye kya ho gaya friends Plz sabhi theatre me ja kr race 3 fir se dekho... kam se kam 3 tickets to kharido Film 300 crores krni chahiye.. plz
array([0])
```

```
# Creating a pipeline
from sklearn.pipeline import Pipeline
pipe3 = Pipeline([('tfidf', TfidfVectorizer(max_features = 270)), ('sgd_model', SGDClassifier())])
pipe3.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer(max_features=270)),
                ('sgd_model', SGDClassifier())])
```

```
z = str(input())
pipe3.predict([z])
```

```
Ye kya ho gaya friends Plz sabhi theatre me ja kr race 3 fir se dekho... kam se kam 3 tickets to kharido Film 300 crores krni chahiye.. plz
array([0])
```

```
# Creating a pipeline
from sklearn.pipeline import Pipeline
pipe4 = Pipeline([('tfidf', TfidfVectorizer(max_features = 270)), ('cll', lgb.LGBMClassifier())])
pipe4.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer(max_features=270)),
                ('cll', LGBMClassifier())])
```

```
w = str(input())
pipe4.predict([w])
```

```
Ye kya ho gaya friends Plz sabhi theatre me ja kr race 3 fir se dekho... kam se kam 3 tickets to kharido Film 300 crores krni chahiye.. plz
array([0])
```

5. Link to our Google Colaboratory Code:

   https://colab.research.google.com/drive/1WLrxIA2cL0XhuiSrLNicb6LpXeokaXmL
   ?usp=sharing

Result Analysis

1. Comparing with Previous Studies:

Table 1. presents the accuracy of the different classifiers done in the previous studies.

Sr. No	Authors	Year	Feature	Classifier	Accuracy	Dataset Acquired From
1	Raisi & Huang[12]	2016	-	-	They did not evaluate their proposed model	Twitter and Ask.fm datasets
2	Reynolds, Kontostathis and Edwards[13]	2011	BoW	SMO, IBK, JRip, J48	The model was capable of recognizing 78.5% posts in Formspring dataset	Formspring Link: www. Formspring.me
3	Nahar et al. [14]	2014	TF-IDF unigrams	Ensemble	-	MySpace, Slashdot, Kongregate, Twitter
4	Yin et al. [15]	2009	TF-IDF	SVM	Kongregate (0.289) Slashdot (0.273) MySpace (0.351)	MySpace, Slashdot, Kongregate
5	Bayzick et al. [16]	2011	Second person pronouns, Insult word, Swear word,	-	Correctly identify 85.3% as cyberbullying posts and 51.91% as innocent posts of MySpace dataset	MySpace
6	Rafiq et al. [17]	2018	Negative comments, Total negative words, Unigrams	AdaBoost, LR	-	Datasets of Vine
7	Galán-García et al. [18]	2016	TF–IDF, N-gram	NB, KNN, RF, J48, SMO	- SMO (68.47%) - J48 (65.81) - RF (66.48%) - NB (33.91%) - KNN (59.79%)	Twitter
8	Al-garadi et al.[19]	2015	Unigram 3-g	SVM, NB	- Naïve Bayes (71%) - SVM (78%)	Twitter
9	Salminen et al. [20]	2020	TI-IDF	LR, NB, SVM, XGBoost	LR (76.8%) - NB (60.6%) -SVM (64.8%) - XGBoost (77.4%)	A total of 197,566 comments from four

						platforms: YouTube, Reddit, Wikipedia, and Twitter,
10	Dinakar et al. [21]	2012	Profanity, BoW, TF-IDF, Weighted unigrams	J48, SVM, NB -based learner, Rule-based Jrip	-NB (63%) - J48 (61%) -SVM (72%) - Rule-based Jrp (70.39%)	Formspring, Youtube
11	Dadvar et al.[22]	2013	Emoticons, Message length, N-gram, Bully keywords, Pronouns	SVM	-	YouTube
12	Van Hee et al[23]	2018	Character n-gram BoW, Word n-gram BoW	LSVM	F1 score of 64% and 61% for English and Dutch respectively	Posts of ASKfm in Dutch and English
13	Cheng et al.[24]	2019	-	LR, LSVM, RF	-	Vine, Instagram
14	Muneer et al.[25]	2020	TF-IDF and Word2Vec	LR, LGBM, SGD, RF, AdaBoost, NB, and SVM	- LR (90.57%) - LGBM (90.55%) - SGD (90.6%) - RF (89.84%) - AdaBoost (89.30%) - NB (81.39%) - SVM (67.13%)	Twitter
15	Our Study	2023	BoW, Tf-Idf	RF,LR,SGD,LGBM	- RF (57%) - SGD (63%) - LGBM (63%) - LR(60%)	Twitter threads, Instagram comments, Youtube comments. Note: All the cyberbullying scripts are targeted for Indians and are collected from Indian users.

Table 1.

2. <u>Performance Summary of all Algorithms</u>

Table 2. & Chart 1. presents the performance summary of the algorithms used in our Cyberbullying Detection Model. We can see that LightGBM and Stochastic Gradient Descent have similar accuracy,recall and F1 score. However, the precision of LightGBM is more than Stochastic Gradient Descent.

Sr. No	Algorithm	Accuracy	Precision	Recall	F1 score
1	Random Forest	57%	56%	57%	56%
2	Logistic Regression	60%	60%	60%	60%
3	Stochastic Gradient Descent	63%	63%	63%	63%
4	Light GBM	63%	64%	63%	63%

Table 2.

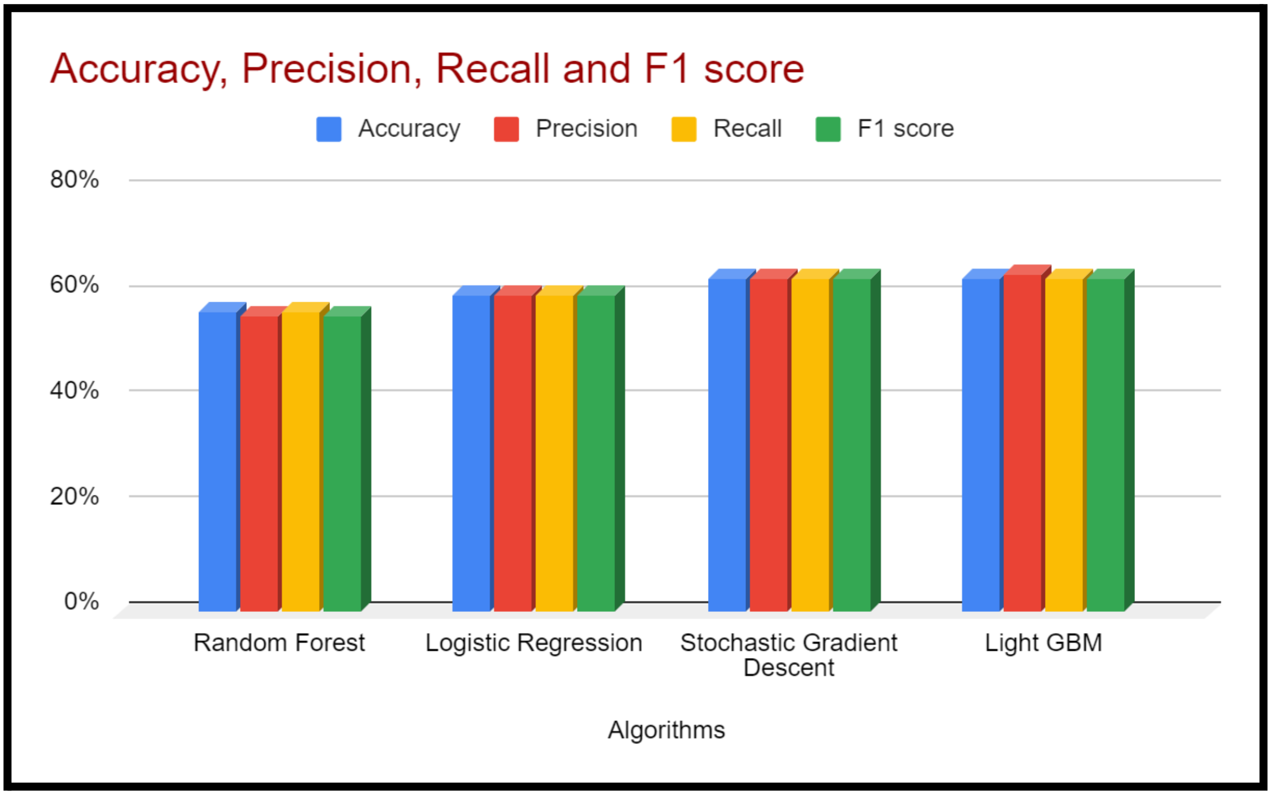


Chart 1. Performance Summary of Algorithms

3. F-measure of all Algorithms

We know that precision helps us check how many of our correctly predicted labels are actually correct. Precision determines the cost of false positives. Recall helps us check if our predictions are actually correct. Recall is used when there is a high cost of true negatives. In order to maintain the balance of Precision and Recall, F-measure is calculated. Table 3, shows SGD & LGBM have equal F-measure.

Sr. No	Algorithm	F-measure
1	Random Forest	56%
2	Logistic Regression	60%
3	Stochastic Gradient Descent	63%
4	Light GBM	63%

Table 3.

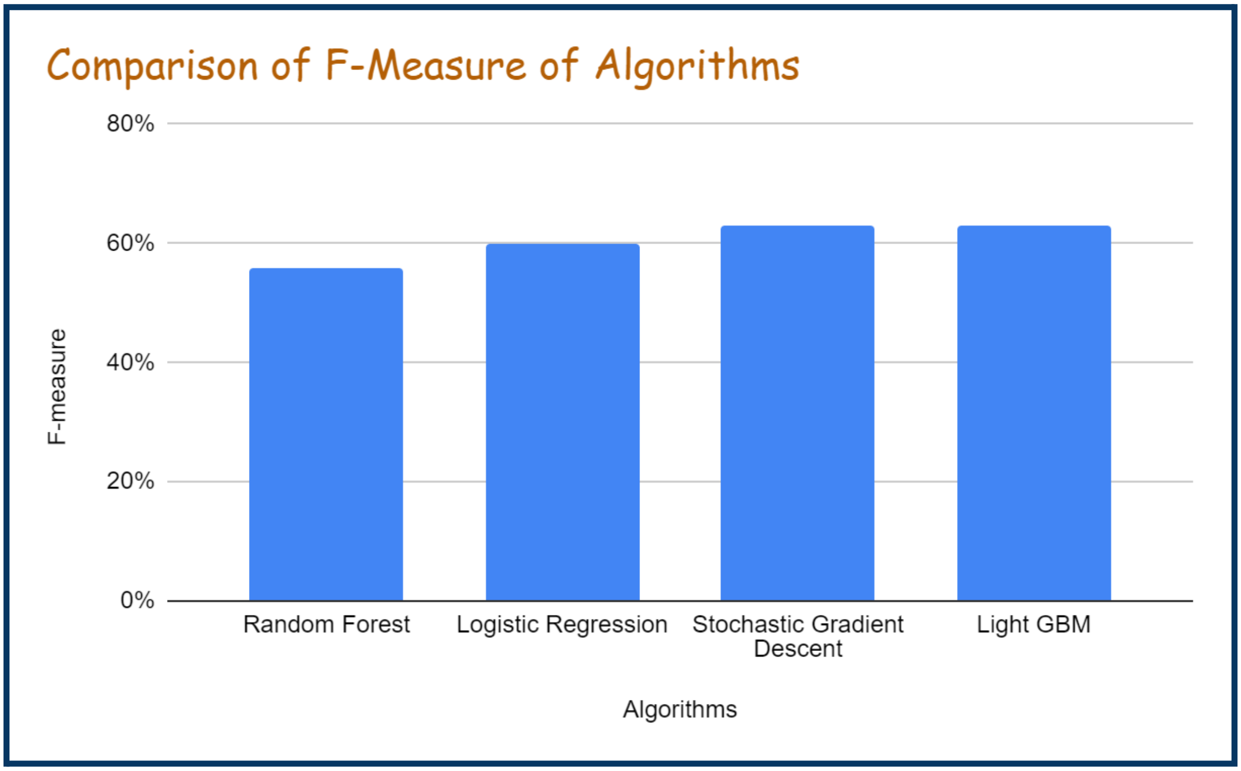


Chart 2. Comparison of Algorithms on the basis of F-measures

4. Test Cases of Algorithms

Sr. No	Script	Label	RF	LR	SGD	LGBM
1	Script 1	1	Correct	Correct	Correct	Correct
2	Script 2	1	Correct	Correct	Correct	Correct
3	Script 3	0	Correct	Correct	Correct	Correct
4	Script 4	1	Correct	Correct	Correct	Wrong
5	Script 5	0	Correct	Correct	Wrong	Correct
6	Script 6	1	Correct	Correct	Correct	Correct
7	Script 7	0	Correct	Wrong	Correct	Correct
8	Script 8	1	Correct	Correct	Correct	Wrong
9	Script 9	1	Correct	Correct	Correct	Correct
10	Script 10	0	Correct	Correct	Correct	Correct

Table 4.

Even though our previous observations suggest that LightGBM and SGD have higher precision,recall,F1-score and accuracy than the other algorithms. However when we used test cases to check the accuracy of algorithms, Random Forest turned out to be the best. LGBM is only 80% correct in the test cases. There are some wrong predictions for SGD and LR. However RF has 100% correct predictions.

5. Time Complexity of Algorithms

Sr. No	Algorithm	Training Time	Prediction Time	Remarks
1.	Random Forest	**07.20 s**	01.39 s	Worst Training Time
2.	Logistic Regression	02.25 s	**01.03 s**	Best Prediction Time
3.	Stochastic Gradient Descent	**01.43 s**	**02.00 s**	Best Training Time & Worst Prediction Time
4.	Light GBM	02.62 s	01.20 s	Average Training Time & Average Prediction Time

Table 5.

Table 5. shows the time complexity of ML Algorithms used in our model. Random Forest takes the most time for training whereas Stochastic Gradient Descent takes the shortest training time. Logistic Regression takes the shortest time for prediction and Stochastic Gradient Descent takes the longest time for prediction.

Overall LightGBM takes average training time and average prediction time.

Conclusion

As the usage of Social Media Platforms is increasing Cyberbullying is becoming more common. As cyberspace can guarantee us complete anonymity and everyone has the right to express their opinions, people feel free to do mis practices such as hurtful speech, harmful comments and abusive trolls. Our Project is mainly focused on Indian audiences and the comments made by Indian audiences to Indian Celebrities,Politicians and Social Media Influencers. We have used 4 classification algorithms namely, Random Forest Classifier, Logistic Regression, Stochastic Gradient Descent and Light GBM. Our dataset was trained on these algorithms. We have also tested the prediction accuracy made by these algorithms. SGD and LightGBM have higher accuracy,precision,recall and F1-score than Random Forest Classifier and Logistic Regression. However, Logistic

Regression has the best prediction time. But SGD has the best training time and average prediction time. Moreover, when we used test scripts among all the 4 algorithms, Random Forest Classifier predicted most correctly. Our future plan includes making a hybrid model which has the prediction time of Logistic Regression, correctness of Random Forest Classifier, accuracy,precision, recall and F1 score of SGD & LightGBM. The performance of our algorithms can also be increased by adding more data to our datasets.

In this way, we have designed a cyberbullying detection model to check whether a script contains cyberbullying content or not.

Works Cited

1. Mishra, Sanjay.
   "Identification and Detection of Cyberbullying on Facebook Using Machine Learning Algorithms."
   Journal of Cases on Information Technology, vol. 23, no. 4, 2021, p. 21. *Online Google Search*, https://orcid.org/0000-0002-3556-9331. Accessed 10 December 2022.

2. Shivashankar, B. S.
   "A Critical Analysis of Cyber Bullying in India-with Special Reference to Bullying in College."
   International Journal of Pure and Applied Mathematics, vol. 119, no. 17, 2018, p. 12. *1314-3395*, http://www.acadpubl.eu/hub/. Accessed 10 December 2022.
3. Venkataraghavan, Manjula.
   "A Study on the Usage of Mobile Phones for Cyber Bullying Among Tweens & Teens of Chennai, India."
   Online Journal of Communication and Media Technologies, vol. 1, no. 1, 2015, p. 12. *Google Scholar*. Accessed 10 December 2022.

4. Dr. Aarti Tolia,
   "Cyberbullying: Psychological Effect on Children", T
   The international journal of Indian Psychology, Vol v3, Issue 2, No.1, Google Scholar.
5. Rounak Ghosh, Siddhartha Nowal, Dr. G. Manju
   "Social Media Cyberbullying Detection using Machine Learning in Bengali Language" *International Journal of Engineering Research & Technology (IJERT) Vol. 10 Issue 05, ISSN:2278-0181*,
   "https://www.ijert.org/social-media-cyberbullying-detection-using-machine-learning-in-bengali-language"

6. Ms. Saurabhi Negi "Indian Journal of Educational Studies : An Interdisciplinary Journal 2016", Vol.3, No.1, ISSN No. 2349-6908

7. Adya Bansal, Akash Baliyan, Akash Yadav, Aman Kamlesh, Hemant Kumar Baranwal "Cyberbullying Detection on Social Networks Using Machine Learning Approaches"
Volume: 09 Issue: 05 | May 2022
Dept. of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut, U.P. India

8. J.I. Sheeba, S. Pradeep Devaneyan , Revathy Cadiravane
"Identification and Classification of Cyberbully Incidents using Bystander Intervention Model"
International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-2S4, July 2019

9. Pavitar Parkash Singh1, Vijay Kumar2, Majid Sadeeq
"Cyber Bullying as an Outcome of Social Media Usage: A Literature Review"
Professor and Associate Dean, 2Associate Professor&Head, 3Ph.D. Research Scholar, School of Education, Lovely Professional University, Phagwara, Punjab, India

10. Shilpi Jain ,Soni Agrawal "Perceived vulnerability of cyberbullying on social networking sites: effects of security measures, addiction and self-disclosure" May 2020

11. Manpreet Kaur, Munish Saini "Indian Government initiatives on Cyberbullying: A case study on cyberbullying in Indian higher education institutions"
Education and Information Technologies | https://doi.org/10.1007/s10639-022-11168-4

12. Raisi, E.; Huang, B. Cyberbullying Identification Using Participant-Vocabulary Consistency. arXiv 2016, arXiv:1606.08084.

13. Reynolds, K.; Kontostathis, A.; Edwards, L. Using Machine Learning to Detect Cyberbullying. In Proceedings of the 2011 10th International Conference on Machine

Learning and Applications and Workshops, Honolulu, HI, USA, 18–21 December 2011; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2011; Volume 2, pp. 241–244.

14. Nahar, V.; Li, X.; Zhang, H.L.; Pang, C. Detecting cyberbullying in social networks using multi-agent system. Web Intell. Agent Syst. Int. J. 2014, 12, 375–388. [CrossRef]

15. Yin, D.; Xue, Z.; Hong, L.; Davison, B.D.; Kontostathis, A.; Edwards, L. Detection of harassment on web 2.0. In Proceedings of the Content Analysis in the WEB, Madrid, Spain, 21 April 2009; Volume 2, pp. 1–7.

16. Bayzick, J.; Kontostathis, A.; Edwards, L. Detecting the Presence of Cyberbullying Using Computer Software. 2011. Available online: https://april-edwards.me/BayzickHonors.pdf (accessed on 28 January 2023).

17. Ibn Rafiq, R.; Hosseinmardi, H.; Han, R.; Lv, Q.; Mishra, S. Scalable and timely detection of cyberbullying in online social networks. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing—SAC ’18, Pau, France, 9–13 April 2018; pp. 1738–1747.

18. Galán-García, P.; De La Puerta, J.G.; Gómez, C.L.; Santos, I.; Bringas, P.G. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. Log. J. IGPL 2015, 24, jzv048. [CrossRef]

19. Al-garadi, M.A.; Varathan, K.D.; Ravana, S.D. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. Comput. Hum. Behav. 2016, 63, 433–443. [CrossRef]

20. Salminen, J.; Hopf, M.; Chowdhury, S.A.; Jung, S.-G.; Almerekhi, H.; Jansen, B.J. Developing an online hate classifier for multiple social media platforms. Hum. Cent. Comput. Inf. Sci. 2020, 10, 1–34. [CrossRef]

21. Dinakar, K.; Reichart, R.; Lieberman, H. Modeling the detection of textual cyberbullying. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.

22. Dadvar, M.; Jong, F.D.; Ordelman, R.; Trieschnigg, D. Improved cyberbullying detection using gender information. In Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012), University of Ghent, Gent, Belgium, 23–24 February 2012.

23. Van Hee, C.; Jacobs, G.; Emmery, C.; Desmet, B.; Lefever, E.; Verhoeven, B.; De Pauw, G.; Daelemans, W.; Hoste, V. Automatic detection of cyberbullying in social media text. PLoS ONE 2018, 13, e0203794. [CrossRef]

24. Cheng, L.; Li, J.; Silva, Y.N.; Hall, D.L.; Liu, H. XBully: Cyberbullying Detection within a Multi-Modal Context. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; ACM: New York, NY, USA, 2019; pp. 339–347

25. Muneer, Amgad. “A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter.” *FutureInterest*, vol. 1, no. 1, 2020, p. 19. *A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter*, A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter.