

CYBERBULLYING TWEET DETECTION

PROJECT DOCUMENTATION

Submitted by

**SANTHOSH PRABHU V
SATYAM PRADHAN
PAMIDI VENKATESH**

As a course requirement in partial fulfilment of

**AICTE CERTIFIED TWO MONTH VIRTUAL INTERNSHIP
IN
MACHINE LEARNING
(DATA SCIENCE WITH PYTHON PROGRAMMING)**



**DATA SCIENCE ACADEMIA, THE DATA SCIENCE WING
SAI CHAMUNDEESWARI ACADEMY, CHENNAI
MSME (UDYAM-TN-02-0166477), GOVT. OF INDIA**

JUNE 2023 TO AUGUST 2023

SAI CHAMUNDEESWARI ACADEMY, CHENNAI

BONA FIDE CERTIFICATE

Certified that this project report titled “**CYBERBULLYING TWEET DETECTION**” is the bona fide work of **SANTHOSH PRABHU V, SATYAM PRADHAN and PAMIDI VENKATESH**, who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein has been done after a good perusal of existing research literature and other resources pertaining to the afore mentioned title of the project and a genuine attempt to better the existing benchmark of the selected dataset.

AJAY MUKUND S, (Ph.D., Anna University)

**B.C.A., M.Sc. Data Science and Computing, PGP- Data Science, GATE –
Computer Science and IT, UGC NET JRF – Computer Science and
Applications, International Young Researcher of the Year 2022**

Project Guide and Supervisor

Internship Trainer and Manager, Data Science Academia

Sai Chamundeeswari Academy

West Tambaram, Chennai – 600 045

ACKNOWLEDGEMENT

We express our heartfelt gratitude to Data Science Academia, the data science wing, Sai Chamundeeswari Academy for providing us with the opportunity to undertake this project as part of the AICTE Certified Two-Month Virtual Internship. The knowledge and skills acquired during this internship will undoubtedly shape our future pursuits in the field of Machine Learning.

We extend our sincere thanks to Mr. Ajay Mukund S, whose guidance, expertise, and continuous support have been instrumental in shaping this project. His extensive knowledge in Data Science and his commitment to mentoring have been a constant source of motivation for us.

We extend our thanks to fellow teammates and interns who helped us in the successful completion of our project Cyberbullying tweet detection. This journey has been a profound learning experience, and we acknowledge the contributions of various individuals without whom this endeavor would not have been possible.

SANTHOSH PRABHU V

SATYAM PRADHAN

PAMIDI VENKATESH

ABSTRACT

The rise of social media has brought about unprecedented connectivity, but it has also given rise to the disturbing phenomenon of cyberbullying, where harmful and offensive content is directed at individuals online. Our goal is to identify and classify different types of cyberbullying tweets with high accuracy, contributing to the creation of safer online environments. This project involves a pipeline encompassing data preprocessing, feature extraction, model implementation, evaluation, and hyperparameter tuning. Using natural language processing techniques, we extract valuable features from textual data using TF-IDF vectorization. Subsequently, we implement a range of classification algorithms, including Logistic Regression, Support Vector Machines, Random Forest, and Naive Bayes. Through an extensive evaluation process, we assess the performance of these models in accurately classifying various forms of cyberbullying. We employ metrics such as accuracy, precision, recall, and F1-score to gauge their effectiveness. Furthermore, we employ hyperparameter tuning using RandomizedSearchCV and GridSearchCV to optimize model configurations and enhance their predictive capabilities. Our findings reveal that, after hyperparameter tuning, our models achieve competitive accuracies, with Logistic Regression and Support Vector Machines leading the pack. Further exploration of ensemble methods, deep learning, and advanced preprocessing techniques will enable us to create safer online spaces by accurately identifying and combating cyberbullying.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	
	1.1 OBJECTIVE AND MOTIVATION	1
	1.2 SCOPE AND APPLICATION	3
	1.3 CHALLENGES	5
2	LITERATURE REVIEW	
	2.1 A COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR CYBERBULLYING DETECTION ON TWITTER	8
	2.2 CYBERBULLYING IDENTIFICATION IN TWITTER USING SUPPORT VECTOR MACHINE AND INFORMATION GAIN BASED FEATURE SELECTION	10
	2.3 CYBERBULLY: AGGRESSIVE TWEETS, BULLY AND BULLY TARGET IDENTIFICATION FROM MULTILINGUAL INDIAN TWEETS	11
	2.4 HATEFUL SYMBOLS OR HATEFUL PEOPLE? PREDICTIVE FEATURES FOR HATE SPEECH DETECTION ON TWITTER	12

	2.5 AUTOMATED HATE SPEECH DETECTION AND THE PROBLEM OF OFFENSIVE LANGUAGE.	13
3	DISCUSSED ML ALGORITHMS	
	3.1 LOGISTIC REGRESSION	14
	3.2 SUPPORT VECTOR MACHINE	15
	3.3 RANDOM FOREST	17
	3.4 NAIVE BAYES	18
	3.5 HYPERPARAMETER OPTIMIZATION	19
	3.5.1 LOGISTIC REGRESSION	20
	3.5.2 SUPPORT VECTOR MACHINE	21
4	EXPLORATORY DATA ANALYSIS	
	4.1 SOURCE OF THE DATASET	23
	4.2 BASIC INFO	24
	4.3 BENCHMARK	26
5	INDIVIDUAL CONTRIBUTIONS	
	5.1 CONTRIBUTION 1	27
	5.2 CONTRIBUTION 2	28
	5.3 CONTRIBUTION 3	39
6	MODEL ARCHITECTURE	
	6.1 DATA PREPROCESSING	29

	6.2 SYSTEM DESIGN - BLOCK DIAGRAM	31
	6.3 CODE IMPLEMENTATION	32
	6.4 RESULTS AND COMPARISON	41
7	CONCLUSION	43
8	FUTURE WORK	44
9	REFERENCES	45

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
2.1	RESULTS OF COMPARATIVE ANALYSIS OF MACHINE LEARNING	10
2.2	RESULTS OF MULTILINGUAL AGGRESSIVE TWEET DETECTION	20
2.3	RESULTS OF PREDICTIVE FEATURES OF HATE SPEECH DETECTION	25

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	FITTING OF LOGISTIC REGRESSION	15
3.2	SVM HYPERPLANE	16
3.3	PREDICTION OF RANDOM FOREST	17
3.4	NAIVE BAYES CLASSIFIER	18
3.5	NAIVE BAYES PROBABILITY	19
4.1	DATASET	24
4.2	DISTRIBUTION OF CLASSES	25
6.1	BLOCK DIAGRAM OF DATA PREPROCESSING	31
6.2	BLOCK DIAGRAM OF CYBERBULLYING DETECTION	32
6.3	IMPORTING THE DATASET	33
6.4	ENCODING THE CLASSES	33
6.5	DATA CLEANING	34
6.6	TF-IDF VECTORIZATION	34
6.7	HYPERPARAMETER OPTIMIZATION USING RANDOM SEARCH	35
6.8	BEST MODEL OF LR RANDOM SEARCH	35

6.9	GRID SEARCH IMPLEMENTATION	36
6.10	GRID SEARCH BEST ELIMINATOR	36
6.11	CONFUSION MATRIX OF BOTH GRID SEARCH AND RANDOM SEARCH SINCE BOTH GAVE EXACT SAME ACCURACY	36
6.12	SVM	37
6.13	SVM RANDOM SEARCH	38
6.14	RANDOM SEARCH RESULTS	38
6.15	CONFUSION MATRIX OF SVM	38
6.16	RANDOM FOREST IMPLEMENTATION	39
6.17	RANDOM FOREST TUNING	40
6.18	CONFUSION MATRIX OF RANDOM FOREST	40
6.19	NAIVE BAYES IMPLEMENTATION	41

LIST OF ABBREVIATIONS

ML	-	Machine learning
SVM	-	Support Vector Machine
LGBM	-	Light Gradient Boosting Machine
SGD	-	Stochastic Gradient Descent
LSTM	-	Long Short Term Memory
NLTK	-	Natural Language Tool Kit

1. INTRODUCTION

Social media platforms have ingrained themselves into our lives in today's linked society, providing a forum for connection, expression, and communication. However, this digital environment also presents difficulties, one of which is the increase in cyberbullying, a kind of online harassment and hostility. The growing use of social media has increased the impact of cyberbullying, which now affects people of all ages, genders, and races.

The "Cyberbullying Tweet Detection" project looks into the world of machine learning to tackle the essential problem of cyberbullying detection in online communication. Our project focuses on the creation and implementation of cutting-edge machine learning models to automatically recognize and categorize tweet content containing cyberbullying.

Our project develops prediction models that examine linguistic patterns and contextual information in tweets using a variety of machine learning approaches, such as Logistic Regression, Support Vector Machines (SVM), Random Forest Classifier, Naïve Bayes. We aim to find latent correlations that can distinguish between hazardous and non-harmful content by leveraging the capability of natural language processing and feature engineering.

1.1 OBJECTIVE AND MOTIVATION

Online communication has completely changed the way we connect, communicate, and interact in the age of fast digitalization. The increasing increase in cyberbullying is one of the most pressing of the difficulties presented by the digital environment. Through social media and other virtual platforms, the world is becoming more interconnected, and as a result, cyberbullying incidents have increased, putting people of all ages, genders, and backgrounds at risk of mental distress and harm. The Cyberbullying Tweet Detection using Machine Learning project finds its use in this area.

Objective:

Our project's core goal is to effectively identify and categorize tweets containing cyberbullying content by utilizing the strength of cutting-edge machine learning algorithms. Our goal is to simultaneously create models that can quickly detect harmful online activities and help create a more secure and civil online community.

A problem brought on by the rise of cyberbullying transcends national borders. By developing prediction models that can distinguish between tweets that perpetuate harm and those that advance online dialogue, we want to meet this challenge head-on. Our goal is to provide platforms, communities, and individuals with the tools necessary to combat cyberbullying in real time, reducing the emotional toll it has on victims. We will accomplish this by leveraging the powers of machine learning.

Motivation:

Our motivation stems from a strong concern for the welfare and mental health of those who use the internet. Online harassment has unintentionally emerged as a dark underbelly of what was once praised as a platform for communication and information sharing on the internet. The impulse to address this issue is inspired by a sincere desire to use technology for good, to combat the detrimental effects of cyberbullying, and to promote an atmosphere of empathy and respect in virtual places.

Our drive is fueled by an awareness of the necessity to address the emotional toll that cyberbullying has on modern life as digital interactions grow more and more fundamental to it. Due to the anonymity provided by internet platforms, people have become more confident to act in ways they might not otherwise. As a result, there has been an increase in psychological trauma, emotional anguish, and even serious consequences for victims. We are driven by the belief that it is our responsibility to step in and offer a technical solution that encourages constructive relationships while reducing negative conduct.

Our motivation is rooted in the idea that technology should be used to improve rather than jeopardize people's lives. We are committed to using machine learning as a force for social change as well as a technical instrument. We hope to create a future in which online environments serve as arenas for personal development, education, and intercultural understanding. We want to enable people, especially young people, to express themselves honestly without worrying about being intimidated.

1.2 SCOPE AND APPLICATION

The project Cyberbullying Tweet Detection using Machine Learning has a broad and complex scope that includes several aspects of machine learning, natural language processing, social responsibility, and technological innovation. The goal of our initiative is to build a safer and more compassionate digital environment; it goes beyond merely implementing technical solutions to engage with the larger social context of online interactions. The development of machine learning models, dataset gathering and preprocessing, model evaluation, and potential real-world application are all included in the project's scope.

The creation and refinement of machine learning models is one of the main facets of the project's scope. We decipher the complexities of Logistic Regression, Support Vector Machines (SVM), and Random Forest Classifier in order to create predictive models that can precisely identify tweets containing cyberbullying content. To ensure that the models reach the maximum levels of accuracy and precision, substantial data preparation, feature engineering, and hyperparameter tuning are required.

The project's aim also includes the development of an extensive dataset that reflects the variety of content that can be found on social media platforms. This dataset contains a variety of frequently occurring themes, subjects, and language constructions from tweets. In order to ensure that the data used for training and testing are reflective of real-world

circumstances while protecting privacy and individual rights, the dataset gathering process incorporates ethical issues.

The research also investigates several approaches to model evaluation, ranging from common accuracy metrics to more subtle techniques like precision, recall, and F1-score. By employing confusion matrices to examine the models' behavior, we may learn about their advantages and shortcomings. The models are not only accurate but also useful in real-world circumstances thanks to this thorough examination.

The Cyberbullying Tweet Detection using Machine Learning project's applications are numerous and extensive, encompassing both online platforms and a larger digital society. The project's results have the potential to significantly advance numerous causes:

1. **Social Media Platforms Enhancement:** The models developed in this project can be seamlessly integrated into social media platforms, providing automated cyberbullying detection and flagging mechanisms. Platforms can use these models to promptly identify harmful content and take appropriate actions, fostering a more respectful and inclusive online community.
2. **Community Moderation:** Online communities, forums, and discussion boards can benefit from the project's outcomes by implementing real-time monitoring systems. These systems can filter out cyberbullying content, thereby creating a safer environment for individuals to express themselves without fear of harassment.
3. **Educational Initiatives:** The project's success can contribute to educational initiatives that raise awareness about cyberbullying and responsible online behavior. By using the models to analyze historical data, educators can gain insights into trends and patterns, enabling them to design targeted interventions.

4. **Government and Regulatory Bodies:** Regulatory bodies concerned with online safety can use the project's outcomes to better understand the dynamics of cyberbullying and formulate effective policies to combat it. The project's insights can aid in creating a regulatory framework that promotes online safety.
5. **Research and Academia:** The project's methodologies and findings can be a valuable resource for researchers and academics working in the fields of machine learning, natural language processing, and social sciences. It can serve as a reference for future studies aiming to explore similar topics.
6. **Global Online Well-Being:** Ultimately, the project's application reaches beyond specific platforms or communities. By curbing cyberbullying, the project contributes to the creation of a more positive and respectful digital space that enhances the mental well-being of internet users across the globe.

1.3 CHALLENGES

Undertaking the Cyberbullying Tweet Detection using Machine Learning project presents a series of intricate challenges that span technical, ethical, and social dimensions. As technology intersects with the complexities of human behavior in online spaces, these challenges shape the project's trajectory and necessitate thoughtful solutions. In this section, we delve into the challenges inherent to the project and how they were addressed.

1. Data Collection and Quality:

Gathering an inclusive and representative dataset that accurately reflects the diverse landscape of cyberbullying is a formidable challenge. Tweets encompass a wide spectrum of themes, languages, and cultural contexts, making data collection a nuanced endeavor. Ensuring that the dataset captures various forms of cyberbullying while avoiding bias is paramount.

To tackle this challenge, a meticulous data collection process was employed. Datasets were sourced from multiple platforms and regions to achieve diversity. Natural language processing techniques were utilized to preprocess and clean the data, ensuring that the models are trained on meaningful and relevant content. Rigorous quality checks were conducted to eliminate noise and irrelevant entries.

2. Anonymity and Privacy:

Handling user-generated content introduces ethical considerations regarding anonymity and privacy. Striking a balance between using authentic data and protecting the identities of individuals contributing to the dataset presents a complex challenge. Ensuring that individuals' rights are respected while building effective models is crucial.

The project addressed this challenge by anonymizing data and adhering to ethical guidelines. User identifiers were removed or replaced with pseudonyms to safeguard privacy. Transparent documentation highlights the steps taken to ensure ethical data handling, showcasing the project's commitment to responsible use of data.

3. Model Complexity and Interpretability:

Creating machine learning models that accurately recognize cyberbullying involves selecting appropriate algorithms and model architectures. However, complex models can be challenging to interpret, especially for non-technical stakeholders. Balancing model

sophistication with interpretability is crucial to ensure that insights are accessible to diverse audiences.

To navigate this challenge, the project chose a combination of models that strike a balance between accuracy and interpretability. Model performance is communicated through clear metrics like accuracy, precision, and recall, making technical aspects comprehensible to a broader audience. Efforts were made to simplify complex concepts and provide real-world examples of model behavior.

4. Linguistic Nuances and Context:

Tweets often contain linguistic nuances, sarcasm, slang, and cultural references that can confound automated analysis. Understanding the intricacies of language and context is pivotal for accurately identifying cyberbullying content, as misinterpretation can lead to false positives or negatives.

The project addressed this challenge by employing Natural Language Processing (NLP) techniques. These techniques enable the models to grasp contextual nuances and sentiment analysis, enhancing their ability to differentiate between harmless banter and harmful content. Training models on a diverse dataset aids in capturing the richness of language variation.

5. Model Generalization:

Ensuring that the trained models generalize well to new and unseen data is a persistent challenge in machine learning projects. Models that perform well on training data may fail to produce similar results in real-world scenarios due to overfitting or biases present in the training data.

To mitigate this challenge, the project utilized cross-validation techniques during model development. The models' performance on both training and validation datasets was meticulously monitored to identify overfitting or bias. The use of diverse data sources and rigorous testing helps ensure that the models generalize effectively.

6. Changing Landscape of Cyberbullying:

The landscape of cyberbullying constantly evolves, with new forms of harassment and harmful behavior emerging. Creating models that remain effective in identifying novel cyberbullying trends is a perpetual challenge. The project must strike a balance between capturing existing patterns and adapting to new ones.

To address this challenge, the project employs dynamic monitoring and continuous model refinement. Regular updates to the model, fueled by new data and evolving understanding of cyberbullying dynamics, ensure that the models remain relevant and effective against emerging trends.

2. LITERATURE REVIEW

In this section, we explored what other people have already found out about cyberbullying and how to deal with it. The summary, results and conclusions of various research papers and articles are discussed in this literature review section.

2.1 A COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR CYBERBULLYING DETECTION ON TWITTER[1]

This paper discusses the issue of cyberbullying on social media platforms, particularly Twitter, and the need for effective detection of cyberbullying tweets. In this

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

paper, the authors explored the cyberbullying issue by compiling a global dataset of 37,373 unique tweets and used seven different machine learning classifiers and built models to detect the cyberbullying tweets.

The proposed approach of the authors is to detect cyberbullying by extracting tweets, classifying the tweets using text analysis techniques based on predefined keywords, and then classifying the tweets as offensive or non-offensive.

The study highlights the importance of detecting cyberbullying without relying on the victims' interactions and provides insights into the performance of different machine learning techniques for this task. They have compared and analysed different machine learning models and evaluated their performance using accuracy, precision, f1 score, recall and prediction time.

Data preprocessing included stopwords removal, punctuation marks removal, special characters removal, stemming as well as spam content removal.

Feature extraction is a critical step for text classification in cyberbullying. But the authors restricted themselves to 2 feature extraction techniques. In the proposed model, they made use of the TF-IDF and Word2Vec feature extraction. Various methods of text classification were investigated in this paper.

The following table shows the results of the models built and compared by the authors of this paper

S.NO	MACHINE LEARNING CLASSIFIER	ACCURACY ACHIEVED
1	Logistic Regression	90.57%
2	LGBM Classifier	90.55%
3	SGD Classifier	90.6%
4	Random Forest	89.84%
5	AdaBoost Classifier	89.30%

6	Multinomial Naive Bayes	81.39%
7	SVM	67.13%

Table 2.1 Results of Comparative Analysis of Machine Learning

The authors of this paper have observed that Logistic Regression performed better as the data size increased and obtained the best prediction time when compared to the other classifiers. The authors also claimed that this comparison analysis helped them to understand the limitations and advantages of ML in text classification models.

2.2 CYBERBULLYING IDENTIFICATION IN TWITTER USING SUPPORT VECTOR MACHINE AND INFORMATION GAIN BASED FEATURE SELECTION[2]

This research paper focuses on using the Support Vector Machine (SVM) method and Information Gain (IG) feature selection to identify cyberbullying in Twitter. This study is a text classification where more data is used, the more features are produced, therefore this research paper also uses Information Gain as feature selection to select features that are relevant to the classification.

Dataset used in this paper consisted of 300 tweets, with half containing bullying and half without bullying. The data were manually labelled by an expert. In the experiment, the data was splitted into 240 tweets as training data and 60 tweets as testing data.

The preprocessing included processes such as tokenizing, filtering, stemming and then term weighting. The features used in this work are BoW with term frequency-inverse document frequency (TF-IDF) as the term weighting method. The features were then used as the input of SVM. Before the classification task, the IG value of each features are calculated and the features with highest IG values are selected to represent the document.

SVM is used to find the best hyperplane that serves as a separator class negative and positive class. There are several SVM kernel functions available. In this research paper kernel function used is SVM Polynomial.

Using SVM, the classification achieved an accuracy of 75%, precision of 70.27%, recall of 86.66%, and f-measure of 77.61%. The IG feature selection method, with a threshold value of 90%, achieved an accuracy of 76.66%, precision of 72.22%, recall of 86.66%, and f-measure of 78.78%. This suggests that IG feature selection effectively selected relevant features and improved the classification performance.

Overall, the results demonstrate that the combination of SVM and IG feature selection can be effective in identifying cyberbullying in Twitter, with both methods achieving high accuracy and recall rates.

2.3 CYBERBULLY: AGGRESSIVE TWEETS, BULLY AND BULLY TARGET IDENTIFICATION FROM MULTILINGUAL INDIAN TWEETS[3]

This paper report focuses on the detection of cyberbullying in multilingual Indian tweets using LSTM-based classifiers. The report discusses the language detection and aggression detection process, patterns in prospective bullies and bully targets, and the impact of reading aggressive tweets on personal behavior.

The model was trained on over 150,000 tweets in Hindi, English, Bengali, and Hinglish, and showed high accuracy in detecting aggression in different languages. The author says that LSTM(Long Short Term Memory) model was chosen for its ability to better understand the text through its memory unit.

Dataset: The Aggression-annotated Corpus of Hindi-English Code-mixed Data is a labeled dataset that includes three categories: Openly Aggressive (OAG), Covertly Aggressive (CAG), and Non-Aggressive (NAG).

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

The preprocessing included removing stop words, removing white spaces, removing usernames, removing punctuation and numbers, removing urls and lowercasing the tweets.

The results of the study showed that the proposed model achieved high accuracy in detecting aggression in multilingual Indian tweets. Additionally, the study observed that the aggressiveness of the feed had an impact on the behavior of users, suggesting that exposure to aggressive content can influence a person's aggression levels.

Language	Precision
English	0.7255
Hindi	0.8427
Bangla	0.8125
Hinglish	0.8735

Table 2.2 Results of Multilingual Aggressive Tweet Detection

The study concluded that the proposed model can effectively detect aggression in multilingual Indian tweets.

2.4 HATEFUL SYMBOLS OR HATEFUL PEOPLE? PREDICTIVE FEATURES FOR HATE SPEECH DETECTION ON TWITTER[4]

This paper is about identifying hate speech on Twitter. The authors made a list of criteria to find hate speech and used it to analyze a dataset of tweets. They looked at different features like character n-grams and gender to see which ones were most effective in detecting hate speech.

The dataset used in this study consisted of 16,914 tweets, of which 3,383 were annotated as containing sexist content and 1,972 were annotated as containing racist content. The remaining tweets (11,559) were not flagged as containing hate speech. The data was

collected by performing a manual search of common slurs and terms related to religious, sexual, gender, and ethnic minorities.

The following table shows the result of this research paper.

	char n-grams	+gender	+gender +loc	word n-grams
F1	73.89	73.93	73.62	64.58
Precision	72.87	72.93	72.58	64.39
Recall	77.75	77.74	77.43	71.93

Table 2.3 Results of Predictive features of Hate speech Detection

The results showed that character n-grams and gender were the best features for detecting hate speech. The study also talked about other research on offensive language and hate speech and concluded that more data and better classification of demographic information is needed.

2.5 AUTOMATED HATE SPEECH DETECTION AND THE PROBLEM OF OFFENSIVE LANGUAGE[5]

This article discusses the challenges of automated hate speech detection on social media, emphasizing the difficulty of distinguishing hate speech from other offensive language. The authors of the article used a list of hate speech words to find tweets that contained those words. They then labeled the tweets as either hate speech, offensive language, or neither.

The dataset used in this study was collected from Twitter using the Twitter API. The authors compiled a hate speech lexicon containing words and phrases identified by internet users as hate speech, obtained from Hatebase.org. They then searched for tweets containing terms from this lexicon, resulting in a sample of tweets from 33,458 Twitter users. The tweets

were labeled into three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech.

The hate speech lexicon used in the study had low precision, with only 5% of tweets being labeled as hate speech by the majority of coders. The majority of tweets were classified as offensive language (76%) and a smaller portion as non-offensive (16.6%). The study concluded that accurately distinguishing between hate speech and offensive language is important, as conflating the two can lead to mislabeling and considering many people as hate speakers.

The authors say that future research on hate speech detection needs to consider the context and the different ways hate speech is used. They also say that it's important to think about the context and social biases when studying hate speech.

3. DISCUSSED ML ALGORITHMS

3.1 LOGISTIC REGRESSION

Logistic Regression is one of the fundamental machine learning classifiers. It is a widely used machine learning classifier. Logistic Regression, commonly used for binary classification, can also be extended to multi-class classification scenarios. In the context of our project, Cyberbullying Tweet Detection with six classes, Logistic Regression serves as a versatile algorithm for categorizing tweets into different cyberbullying types.

At its core, Logistic Regression models the probability that a given input belongs to a particular class using the logistic function. In our project, we can relate this concept to the classification of tweets. The input features could include parameters such as the presence of certain keywords, linguistic patterns indicative of cyberbullying. The logistic regression model then calculates the probability that the tweet belongs to the cyberbullying class based on these features.

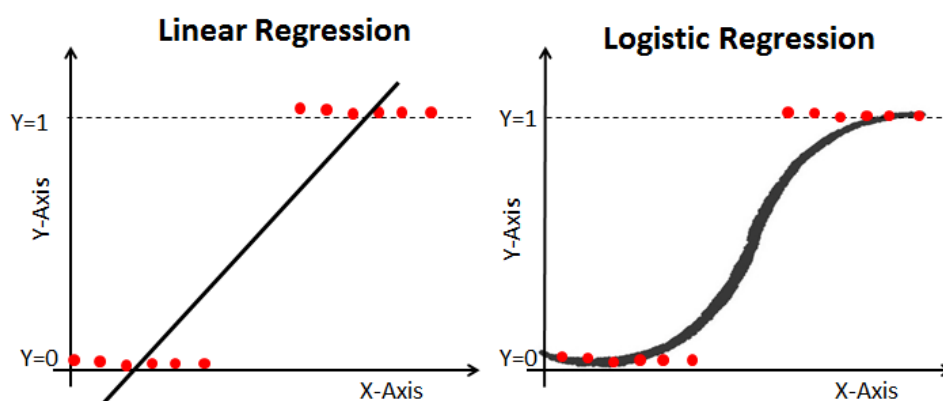


Figure 3.1 Fitting of Logistic Regression Source: analyticsvidhya.com

Multi-class Logistic Regression builds upon the concepts of binary Logistic Regression but adapts them to handle more than two classes. In our case, the algorithm calculates the probability that a given tweet belongs to each of the six cyberbullying classes. It then assigns the tweet to the class with the highest probability.

Logistic Regression is computationally efficient and can handle large datasets without excessive computational demands, which is crucial when working with extensive social media data. Multi-class Logistic Regression can flexibly accommodate more than two classes, making it suitable for our project's six-class cyberbullying classification.

In the context of our project, Logistic Regression is a potent technique to predict whether a given tweet contains elements of cyberbullying. By relating parameters such as age, ethnicity, gender, religion, and other linguistic features to the logistic function, we can create a predictive model that assists in early detection and prevention of cyberbullying instances on social media platforms.

3.2 SUPPORT VECTOR MACHINE

In the realm of machine learning, Support Vector Machines (SVM) stand as a formidable tool for classification and regression tasks. In the Cyberbullying Tweet Detection using Machine Learning project, SVM emerges as a powerful ally in identifying and categorizing cyberbullying content within tweets. The essence of SVM lies in its ability to find the optimal decision boundary, or hyperplane, that effectively separates data points of different classes.

At its core, SVM operates on the principle of maximizing the margin between the hyperplane and the nearest data points from each class. These data points, known as support vectors, dictate the positioning of the hyperplane and influence its ability to generalize to new, unseen data. The primary objective of SVM is to find the hyperplane that best divides the data while minimizing the risk of misclassification.

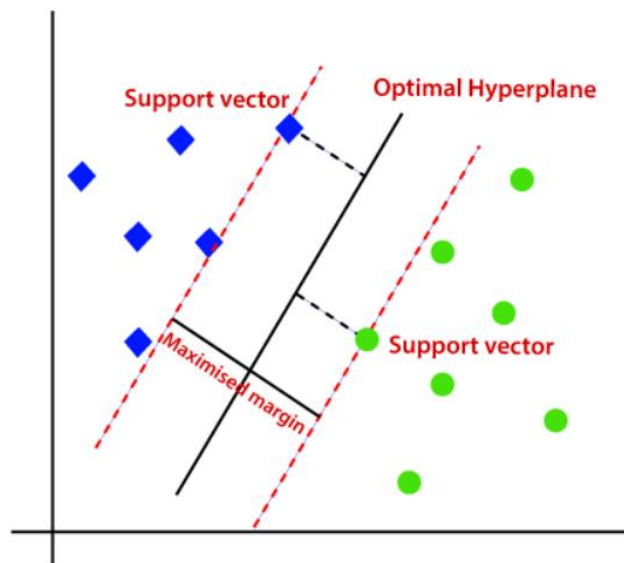


Figure 3.2 SVM Hyperplane Source: medium.com

Kernel Trick:

SVM's power extends beyond linearly separable data. Thanks to the kernel trick, SVM can transcend the constraints of linear boundaries and handle complex, non-linear relationships within the data. This trick involves mapping the original data points into a higher-dimensional space where they can be linearly separated. Kernels, such as the linear kernel and the radial basis function (RBF) kernel, define the nature of this mapping.

In the context of the project, where textual data in the form of tweets is being analyzed, the kernel trick is particularly valuable. It allows SVM to capture intricate linguistic nuances, such as sarcasm, sentiment, and context, that contribute to the differentiation between cyberbullying and non-cyberbullying content.

3.3 RANDOM FOREST

Random Forest, a powerful ensemble learning technique, is well-suited for multi-class classification tasks, such as our Cyberbullying tweet detection project with six classes. This versatile algorithm can effectively analyze various features to categorize tweets into different cyberbullying types.

Random Forest builds a forest of decision trees, each trained on different subsets of the dataset. For multi-class classification, the algorithm combines the outputs of individual trees to determine the final class assignment for a given tweet.

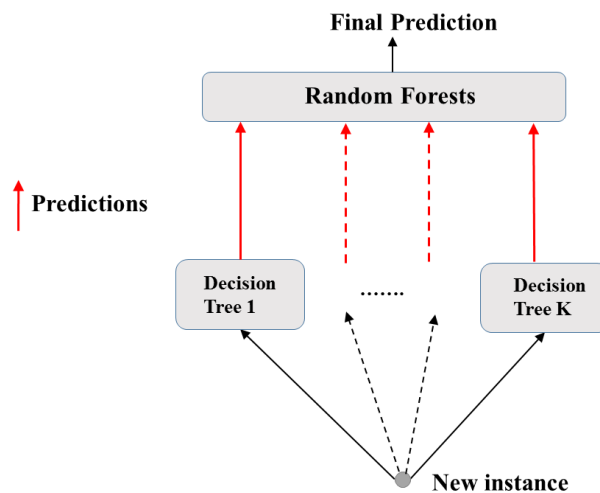


Figure 3.3 Prediction of Random Forest Source: datacamp.com

Random Forest is capable of handling complex relationships between features and classes, leading to accurate multi-class predictions. The algorithm provides insights into feature importance, aiding in understanding the most influential factors driving each class prediction. The aggregation of multiple decision trees in the ensemble helps mitigate overfitting and enhances generalization to new data.

Random Forest stands as a robust and versatile choice for multi-class classification in our project. By harnessing the collective insights of decision trees, this algorithm adeptly categorizes tweets into different cyberbullying types. Through its ability to capture intricate relationships in the data, Random Forest supports the development of proactive strategies to address diverse forms of cyberbullying.

3.4 NAIVE BAYES

Naive Bayes, a probabilistic machine learning technique, is well-suited for multi-class classification tasks like our Cyberbullying tweet detection project with six classes. This algorithm uses probability and assumptions to categorize tweets into different cyberbullying types. Naive Bayes relies on Bayes' theorem and a naive assumption of feature

independence. For multi-class classification, it calculates the probability of a tweet belonging to each class and selects the class with the highest probability as the prediction.

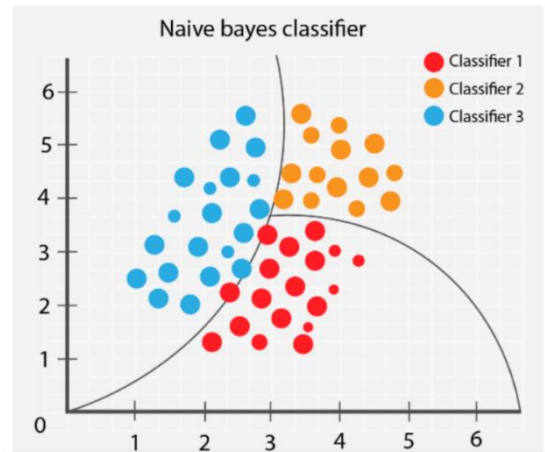


Figure 3.4 Naive Bayes Classifier Source: analyticsvidhya.com

Naive Bayes is straightforward to implement and computationally efficient, making it suitable for quick classification tasks. The algorithm provides probabilities for each class, helping us understand the confidence level of predictions. Naive Bayes can perform well even with a limited amount of training data.

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability

Posterior Probability
Predictor Prior Probability

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Figure 3.5 Naive Bayes Probability Source: medium.com

Naive Bayes offers an intuitive and efficient approach for multi-class classification in our project. By leveraging probability and making reasonable assumptions, this algorithm

effectively categorizes tweets into various cyberbullying types. Its simplicity and interpretability contribute to a proactive approach in addressing different forms of cyberbullying.

3.5 HYPERPARAMETER OPTIMIZATION

Hyperparameter optimization enables us to strike the right balance between underfitting (too simplistic) and overfitting (too complex) our algorithms. By finding the optimal hyperparameters, we ensure that our models generalize well to new, unseen tweets. This ensures that our system is effective in detecting different forms of cyberbullying across various contexts.

This section of our project documentation delves into the methodology behind hyperparameter optimization and its direct impact on the effectiveness of our cyberbullying detection system. Optimization techniques used are Random search and Grid search. Due to time constraint we restricted ourselves to optimization of LR and SVM classifiers.

Random Search offers a balanced approach between exhaustively testing all possible combinations (which can be impractical) and randomly selecting a single configuration (which might not yield good results). By randomly sampling, we explore a variety of hyperparameter settings, increasing the likelihood of finding combinations that significantly enhance the performance of our models.

Grid Search presents a systematic solution that strikes a balance between exhaustively evaluating every hyperparameter combination (a time-consuming endeavor) and the potential pitfalls of selecting a single configuration. By creating a structured grid of predefined hyperparameter values, we traverse this landscape, evaluating each configuration.

3.5.1 LOGISTIC REGRESSION

This section explores how fine-tuning the Logistic Regression model's hyperparameters can significantly enhance its performance in identifying different forms of cyberbullying.

Hyperparameters Under Consideration:

1. Choice of Solver: The solver determines the optimization algorithm used by Logistic Regression. Three options, including 'newton-cg,' 'lbfgs,' and 'liblinear,' are explored to identify the most effective solver for our task.

2. Penalty: The 'l2' penalty encourages Logistic Regression to avoid extreme weight values. This regularization technique prevents overfitting and improves the model's generalization capabilities.

3. Regularization Parameter (C): Denoted as C, the regularization parameter balances the trade-off between fitting the training data well and avoiding overfitting. Different values of C, such as 100, 10, 1.0, 0.1, and 0.01, are investigated.

4. Maximum Iterations: The 'max_iter' parameter defines the maximum number of iterations for the solver to converge. Variations like 1500, 2500, and 5000 are explored to determine the convergence threshold.

5. Fit Intercept: The 'fit_intercept' parameter decides whether or not to calculate the intercept. Both True and False options are assessed to determine the model's sensitivity to intercept presence.

Achieving Optimal Performance:

The convergence of strategic hyperparameter tuning through grid search and cross-validation cultivates an optimized Logistic Regression model. This model is finely attuned

to distinguish between various cyberbullying types within tweets. By leveraging these techniques, our project attains a classification tool that is both powerful and reliable.

Logistic Regression emerges as a potent force in our mission to combat cyberbullying. Through hyperparameter optimization and meticulous tuning, we harness its capabilities to identify harmful content within tweets accurately.

3.5.2 SUPPORT VECTOR MACHINE

This section explores how fine-tuning the SVM model's hyperparameters can significantly enhance its performance in identifying different forms of cyberbullying.

Hyperparameters Under Consideration:

1. Choice of Kernel: The choice of kernel plays a pivotal role in SVM's performance. The linear kernel is effective when the data is linearly separable, while the RBF kernel is suitable for capturing complex non-linear relationships. In the project, both kernels are explored to identify the most suitable one for the task at hand.

2. Regularization Parameter (C) : The regularization parameter, denoted as C, controls the trade-off between maximizing the margin and minimizing the classification error. A smaller C emphasizes a wider margin but allows some misclassification, while a larger C prioritizes accurate classification but might result in a narrower margin. Balancing this trade-off is crucial for avoiding overfitting or underfitting.

3. Gamma for Non-Linear Kernels: For non-linear kernels like RBF, the gamma parameter determines the extent to which a single training example influences the decision boundary. Higher values of gamma result in a more complex decision boundary that closely follows the training data, which might lead to overfitting.

Stratified K-Fold Cross-Validation:

To ensure that the trained SVM model generalizes well to new, unseen data, the project employs stratified k-fold cross-validation. In this technique, the dataset is divided into k subsets, or folds, while maintaining the same class distribution as the original dataset. The model is trained on k-1 folds and validated on the remaining fold. This approach provides a more accurate estimate of the model's performance and helps prevent overfitting.

Achieving Optimal Performance:

By tuning the kernel, regularization parameter, and gamma, the SVM model's performance is optimized. The combination of grid search, random search, and cross-validation ensures that the model is robust, accurately identifying cyberbullying content while avoiding misclassification errors. In conclusion, Support Vector Machines (SVM) play a pivotal role in the "Cyberbullying Tweet Detection using Machine Learning" project. Their capability to handle non-linear data through the kernel trick, coupled with meticulous hyperparameter tuning, results in a model that can effectively distinguish between harmful and harmless content within tweets. The combination of technical expertise and strategic tuning ensures that SVM fulfills its potential as a powerful and accurate classification tool in this essential project.

4. EXPLORATORY DATA ANALYSIS

The Exploratory Data Analysis (EDA) section provides an overview of the data's source and composition, identify significant patterns, detect potential outliers, and explore correlations among variables. This analytical phase serves as a foundational step for the development of cyberbullying tweet detection solution.

4.1 SOURCE OF THE DATASET

In this section, we outline the source of our dataset, which we obtained from Kaggle, a widely recognized platform for sharing and accessing datasets. We are extremely grateful to Kaggle for providing us with dataset for our project and we acknowledge the authors of the dataset.

Authors: J. Wang, K. Fu, C.T. Lu, “SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection,” Proceedings of the 2020 IEEE International Conference on Big Data (IEEE BigData 2020), December 10-13, 2020.

Link to the dataset: <https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification>

4.2 BASIC INFO

Dataset consists of two columns: tweet_text and cyberbullying_type .This dataset contains 47692 tweets and cyberbullying_type is labelled according to the class of cyberbullying:

- Age
- Ethnicity
- Gender
- Religion
- Other type of cyberbullying
- Not cyberbullying

The data has been balanced in order to contain ~8000 of each class. As mentioned by the authors in Kaggle, these tweets either describe a bullying event or are the offense themselves, therefore explore it to the point where you feel comfortable.

1	tweet_text	cyberbullying_type
2	In other words #katandandre, y	not_cyberbullying
3	Why is #aussietv so white? #MK	not_cyberbullying
4	@XochitlSuckkks a classy whore	not_cyberbullying
5	@Jason_Gio meh. :P thanks for	not_cyberbullying
6	@RudhoeEnglish This is an ISIS	not_cyberbullying
7	@Raja5aab @Quickieleaks Yes,	not_cyberbullying
8	Itu sekolah ya bukan tempat bul	not_cyberbullying
9	Karma. I hope it bites Kat on the	not_cyberbullying
10	@stockputout everything but m	not_cyberbullying
11	Rebecca Black Drops Out of Sch	not_cyberbullying
12	@Jord_Is_Dead http://t.co/UsC	not_cyberbullying
13	The Bully flushes on KD http://t	not_cyberbullying
14	Ughhhh #MKR	not_cyberbullying
15	RT @Kurdsnews: Turkish state h	not_cyberbullying
16	Love that the best response to t	not_cyberbullying
17	@yasmimcaci @Bferrarii PAREN	not_cyberbullying

Figure 4.1 Dataset

The dataset consists of 0 null values which reduces the overhead of handling the null values.

The classes were encoded as follows:

- not_cyberbullying: 0
- gender: 1
- religion: 2
- other_cyberbullying: 3
- age: 4
- ethnicity: 5

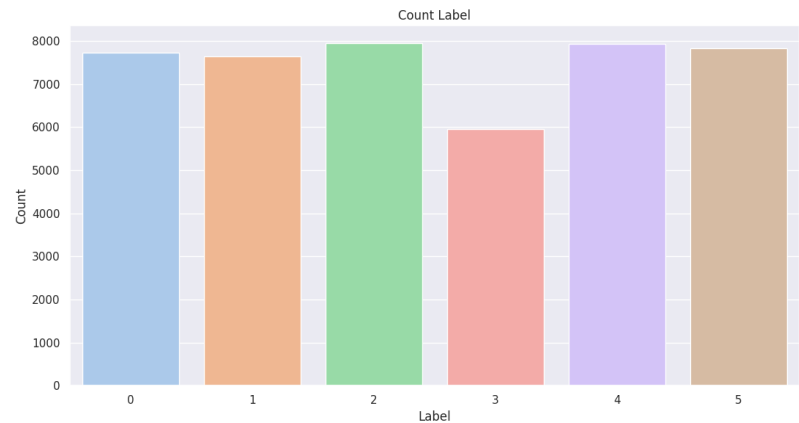
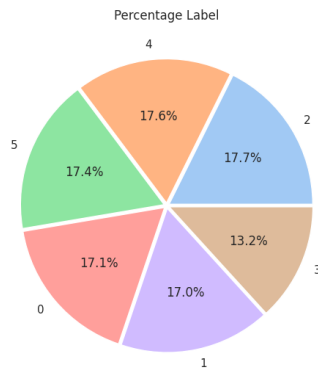


Figure 4.2 Distribution of classes

Cyberbullying type classes description:

- **Age:** This class refers to instances of cyberbullying that revolve around someone's age. It could involve making fun of someone's age, or using age-related insults to hurt them.
- **Ethnicity:** This class refers to tweets that involves bad words.
- **Gender:** Involves attacking someone based on their gender.
- **Religion:** Cyberbullying that is focused on a person's religion or beliefs.
- **Other Type of Cyberbullying:** This class covers any form of online harassment that doesn't fit the defined categories but still causes harm and distress.
- **Not Cyberbullying:** Situations where the content or behavior does not meet the criteria for cyberbullying.

Understanding these different classes helps in identifying the nature of cyberbullying instances and enables effective intervention and prevention strategies.

4.3 BENCHMARK

In our pursuit of developing an accurate Cyberbullying tweet detection model, it's imperative to establish a comprehensive benchmark against which we can assess the effectiveness of our models. This section delves into benchmarking by comparing the

performance of our models with various existing solutions, shedding light on their capabilities and identifying areas for improvement.

Author of Literature Survey: "A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter"

1. Logistic Regression: 90.57%
2. LGBM Classifier: 90.55%
3. SGD Classifier: 90.6%
4. Random Forest: 89.84%
5. AdaBoost Classifier: 89.30%
6. Multinomial NB: 81.39%
7. SVM: 67.13%

Author: Jatan Varma (Kaggle Solution)

1. Logistic Regression (LR): 82.28%
2. Support Vector Machine (SVM): 82.20%
3. Naive Bayes (NB): 67.08%
4. Random Forest (RF): 82.96%
5. SVM (After Tuning): 82.44%

Author: Phonphrm Thawatdamrongkit (Kaggle Solution)

1. RF with CountVectorizer: 79.42%
2. RF with TF-IDF Vectorizer: 80.74%

Author: Jhon Vincent Gupo (Kaggle Solution)

1. Naive Bayes (NB) after Oversampling: 85%
2. Random Forest (RF) after Oversampling: 94%
3. K-Nearest Neighbors (KNN) after Oversampling: 79%

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

Our goal is to develop a robust cyberbullying tweet detection system that surpasses the benchmark set by these existing solutions. By implementing advanced preprocessing techniques, optimizing hyperparameters, and exploring ensemble methods such as Random Forest and Naive Bayes, we aim to achieve enhanced results and contribute significantly to the field of cyberbullying detection.

5. INDIVIDUAL CONTRIBUTIONS

5.1 CONTRIBUTION 1

Team leader: Santhosh prabhu V

- **Dataset Selection:** Curated the cyberbullying tweet dataset from Kaggle to provide a robust foundation for our project.
- **Data Preprocessing:** Led the charge in cleaning and transforming the raw text data, making it suitable for analysis.
- **Algorithm Implementation:** Implementation of Logistic Regression and Naïve Bayes classifiers.
- **Hyperparameter Tuning:** Explored and optimized hyperparameters for the Logistic Regression model to achieve optimal performance.
- **Documentation Preparation:** Took charge of compiling and organizing the project documentation, ensuring clarity and coherence.
- **Result Tracking:** Diligently recorded and analyzed all experimental results, guiding decision-making throughout the project.
- **Literature Survey:** Conducted thorough research, referencing key papers like "A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter," "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," and "Automated Hate Speech Detection and the Problem of Offensive Language."

5.2 CONTRIBUTION 2

Team Member 1: Satyam Pradhan

- Literature Review: Conducted an extensive literature survey, gathering insights from papers like "Cyberbullying Identification in Twitter Using Support Vector Machine and Information Gain Based Feature Selection."
- SVM Model Implementation: Took the lead in building the Support Vector Machine model for our project, leveraging the acquired literature knowledge.
- Hyperparameter Tuning: Conducted thorough hyperparameter tuning for the SVM model, optimizing its performance.
- Collaborative Work: Worked closely with other team members to integrate the SVM model into the overall project framework.

5.3 CONTRIBUTION 3

Team Member 2: Pamidi Venkatesh

- Random Forest Implementation: Led the implementation of the Random Forest classifier, a crucial aspect of our project's classification pipeline.
- Contributions to Logistic Regression: Made significant contributions to the implementation of Logistic Regression, enriching the model's versatility.
- Literature Exploration: Engaged with relevant literature, referencing papers like "Cyberbully: Aggressive Tweets, Bully and Bully Target Identification from Multilingual Indian Tweets" to gain insights.
- Collaborative Engagement: Collaborated actively with other team members to ensure seamless integration of different classifiers within the project.

6. MODEL ARCHITECTURE

6.1 DATA PREPROCESSING

Text preprocessing is a vital step to transform raw text data into a suitable format for analysis. For our Cyberbullying tweet detection project, ensuring that the text within tweets is clean and structured plays a pivotal role in enhancing the accuracy and effectiveness of our models. This section delves into the preprocessing pipeline that prepares the tweets for analysis, employing a range of techniques to refine the text data.

- **Cleaning the Text:**

The first stage involves removing noise from the text, which includes eliminating hashtags, mentions, URLs, and retweets. This is achieved through a regular expression pattern that systematically replaces these elements with blank spaces. Removing these artifacts ensures that our analysis focuses solely on the text content of the tweets.

- **Consolidating Spaces:**

Often, text data contains multiple spaces between words. Such irregularities can complicate analysis and introduce unnecessary variance. By employing a simple technique to split and rejoin the text using single spaces, we ensure a uniform structure that facilitates subsequent processing steps.

- **Lemmatization:**

To reduce words to their base or dictionary forms, we employ lemmatization. Words often appear in various forms due to tense, plurals, and other linguistic variations. By applying the WordNetLemmatizer, we normalize words to their root form, which ensures consistency in subsequent analyses.

- **Emoji Transformation:**

Emojis are an integral part of modern communication, conveying emotions that text alone might not capture. Rather than removing emojis, we convert them to text equivalents and integrate them back into the text. This enables our analysis to interpret and understand the emotional context conveyed by emojis.

- **Eliminating Stopwords:**

Stopwords are frequently occurring words in a language (e.g., "a," "the," "is") that often add little contextual meaning. By removing stopwords from our text, we streamline the data and eliminate distractions that could skew the analysis.

- **Special Character Removal:**

Lastly, we strip the text of any remaining special characters, retaining only alphabetic characters and spaces. This final touch ensures that the text is as clean and concise as possible, ready for subsequent analysis.

By implementing a range of techniques to eliminate noise, standardize expressions, and enhance readability, our preprocessing pipeline sets the foundation for accurate and insightful cyberbullying tweet analysis. This comprehensive approach ensures that our analysis models can focus on the essence of the text, enabling us to effectively identify and combat cyberbullying.

6.2 SYSTEM DESIGN - BLOCK DIAGRAM

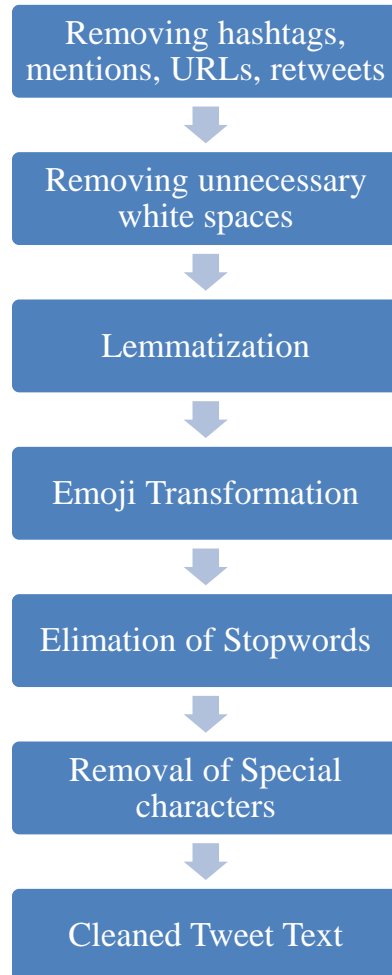


Figure 6.1 Block diagram of Data Preprocessing

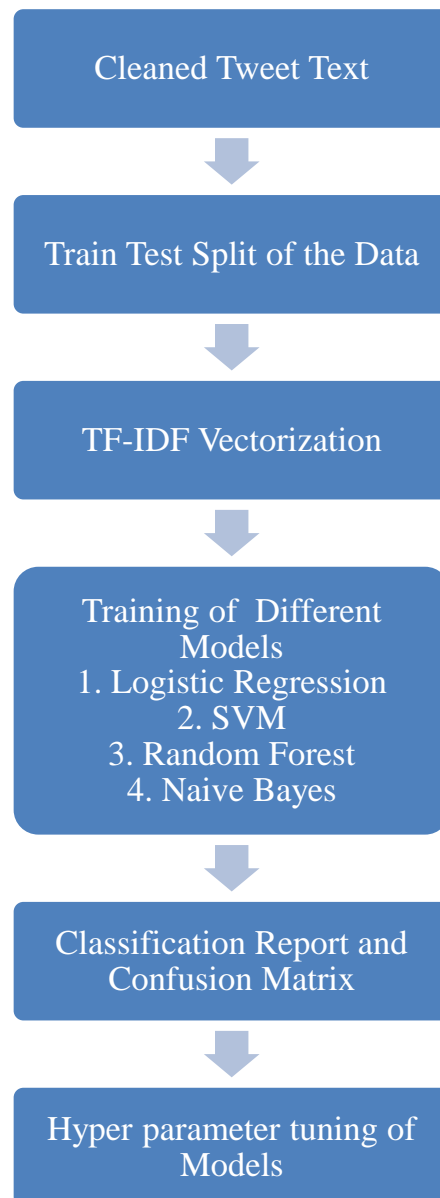


Figure 6.2 Block diagram of Cyberbullying Detection

6.3 CODE IMPLEMENTATION

Data Preprocessing and Exploration: In this section, the necessary libraries are imported, and the required packages are installed. The dataset is loaded and its first few rows are displayed to get an overview.

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

```
[ ] dataset.head()
```

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying

Figure 6.3 Importing the dataset

Data Cleaning and Encoding: Here, duplicates in the dataset are removed based on the cleaned_tweet_text column. This step ensures that each tweet text is unique in the dataset. The "cyberbullying_type" column is then encoded using a predefined dictionary, which maps different types of cyberbullying to numerical values for classification.

Encoding cyberbullying_type in the dataset

```
[ ] ENCODE_DICT = {'not_cyberbullying': 0,
                  'gender': 1,
                  'religion': 2,
                  'other_cyberbullying': 3,
                  'age': 4,
                  'ethnicity': 5}
dataset['cyberbullying_type'] = dataset.cyberbullying_type.replace(ENCODE_DICT)
print(dataset.cyberbullying_type.unique())
```

```
[0 1 2 3 4 5]
```

```
[ ] dataset.sample(5)
```

	tweet_text	cyberbullying_type
41148	@crvtmin – first impression: ion remember – yo...	5
8977	Who's going to be the feminazi of Northmor whe...	1
40424	so the black guy is racist because he got offe...	5
4767	@anggarasuwahju oits... Siapp, blm lengkap. Ha...	0
22807	@SirajZarook @OdinialnVictus @BilaliGhumman @I...	2

Figure 6.4 Encoding the classes

Text Cleaning: This stage removes hashtags, mentions, URLs, retweets, and other unwanted elements from the text. Additionally, it performs lemmatization to reduce words to their base form, converts emojis into text, expands contractions, removes stopwords, and eliminates special characters.

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

```
def cleantext(text):

    #Removal of hashtags, mentions, urls and retweets
    pattern = re.compile(r"#[A-Za-z0-9]+|@[A-Za-z0-9]+|https?:\/\/\S+|www\.\S+|\.[a-z]+|RT @")
    text = pattern.sub('', text)

    #Removal of multiple spaces between words and rejoining using single space
    text = " ".join(text.split())

    #Lemmatize each word in the tweet
    #Lemmatization is the process of reducing words to their base or dictionary form, called the lemma.
    lemma = WordNetLemmatizer()
    text = " ".join([lemma.lemmatize(word) for word in text.split()])

    #Instead of removing the emoji, we convert the emoji in the tweet to text and add it back to the tweet
    emoji = demoji.findall(text)
    for emot in emoji:
        text = re.sub(r"(%s)" % (emot), "_.join(emoji[emot].split()), text)

    #Expansion of contractions
    contractions = {
        r"can't": "can not",
        r"n't": " not",
        r"'re": " are",
        r"'s": " is",
        r"'d": " would",
        r"'ll": " will",
        r"'ve": " have",
        r"'m": " am"
    }
    for contraction, expansion in contractions.items():
        text = re.sub(contraction, expansion, text)

    #Stopwords are commonly used words (e.g., "a", "the", "is") that do not carry significant meaning. Thus stopwords are removed.
    STOPWORDS = set(stopwords.words('english'))
    text = " ".join([word for word in str(text).split() if word not in STOPWORDS])

    #Remove special characters keep only text
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    return text
```

Figure 6.5 Data cleaning

Feature Extraction using TF-IDF: The dataset is split into training and testing sets. The "cleaned_tweet_text" column serves as the feature ("X") while the "cyberbullying_type" column is the target ("y"). Then, the TF-IDF vectorization technique is applied using TfidfVectorizer to transform the text data into numerical representations. This creates a matrix where each row corresponds to a document, and each column corresponds to a unique word in the text.

```
vectorizer = TfidfVectorizer()

X_train_tf = vectorizer.fit_transform(X_train)
X_test_tf = vectorizer.transform(X_test)

print(X_train_tf.shape)
print(X_test_tf.shape)

(36064, 39774)
(9016, 39774)
```

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

Figure 6.6 TF-IDF Vectorization

Logistic Regression Model and Evaluation: A logistic regression model is created. The `max_iter` parameter is set to ensure the optimization algorithm converges. The model is trained on the TF-IDF transformed training data using the `fit()` function. Predictions are made on the test data using the `predict()` function.

The confusion matrix is generated to visualize the model's performance in classifying different types of cyberbullying. Finally, a classification report is generated to provide precision, recall, and F1-score metrics for each class.

A dictionary `param_dist` is defined, containing the hyperparameters to be tuned: `solver`, `penalty`, `C`, `max_iter`, and `fit_intercept` is created and tuning is done using `Randomsearch` and `Gridsearch`.

```
random_search = RandomizedSearchCV(estimator=logreg, param_distributions=param_dist,
                                   n_iter=50, cv=5, scoring='accuracy', n_jobs=-1)

random_search.fit(X_train_tf, y_train)
```

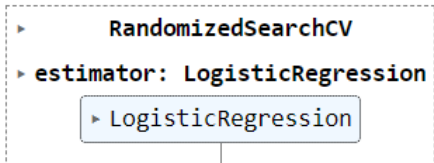


Figure 6.7 Hyperparameter optimization using Random Search

```
best_lr_model.fit(X_train_tf, y_train)
```

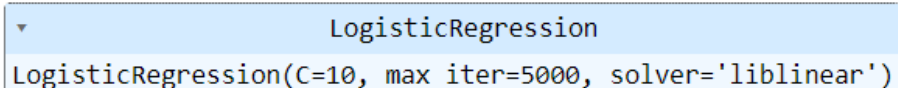


Figure 6.8 Best model of LR Random search

```

LR = LogisticRegression(max_iter=5000)
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values = [100, 10, 1.0, 0.1, 0.01]
iterations = [1500, 2500, 5000]
intercept = [True, False]

grid = dict(solver=solvers,penalty=penalty,C=c_values,max_iter=iterations,
            fit_intercept=intercept)

cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=42)
grid_search = GridSearchCV(estimator=LR, param_grid=grid, n_jobs=-1,
                           cv=cv, scoring='accuracy', error_score=0)
grid_result = grid_search.fit(X_train_tf, y_train)

```

Figure 6.9 Grid Search Implementation

```
grid_result.best_estimator_
```

```

LogisticRegression
LogisticRegression(C=10, max_iter=1500, solver='liblinear')

```

Figure 6.10 Grid Search Best Estimator

Confusion Matrix

	0	1	2	3	4	5
Actual 0	1001	97	64	356	73	18
1	138	1252	9	91	7	9
2	48	3	1514	16	1	1
3	371	65	15	659	23	16
4	35	1	2	12	1560	1
5	14	3	1	12	1	1527
	0	1	2	3	4	5
	Predicted					

Figure 6.11 Confusion matrix of both Grid Search and Random Search since both gave exact same accuracy

Support Vector Machine (SVM) Model and Evaluation: A Support Vector Machine model is constructed for cyberbullying classification. The model aims to classify various types of cyberbullying by learning patterns from the training data. The C parameter, representing the regularization strength, is adjusted to influence the balance between achieving a low training error and a low testing error. The kernel parameter determines the type of decision boundary created by the SVM, whether linear (linear) or more complex (rbf). The gamma parameter controls the curvature of the decision boundary, influencing how closely the model fits the training data. Additionally, the class_weight parameter addresses class imbalances by assigning different weights to different classes during training.

```
from sklearn.svm import SVC

svm = SVC()

svm.fit(X_train_tf, y_train)

▼ SVC
SVC()

y_pred_svm = svm.predict(X_test_tf)

accuracy_svm = accuracy_score(y_test, y_pred_svm)
print("Accuracy:", accuracy_svm)

Accuracy: 0.8312999112688554
```

Figure 6.12 SVM Model

The SVM model is trained on the training data transformed using the TF-IDF vectorization through the fit() function. Subsequently, predictions are made on the test data using the predict() function. To visualize the model's performance, a confusion matrix is generated. This matrix provides insights into how well the model classified different categories of cyberbullying.

```
random_search_svm = RandomizedSearchCV(svm, param_distributions=param_dist,
                                       n_iter=10, cv=stratified_kfold, random_state=42, n_jobs=-1)

random_search_svm.fit(X_train_tf, y_train)
```

```
► RandomizedSearchCV
  ► estimator: SVC
    ► SVC
```

Figure 6.13 SVM Random Search

```
best_params = random_search_svm.best_params_
best_model = random_search_svm.best_estimator_

y_pred_svm = best_model.predict(X_test_tf)

rs_accuracy_svm = accuracy_score(y_test, y_pred_svm)
print("Accuracy:", rs_accuracy_svm)

Accuracy: 0.8317435669920142
```

Figure 6.14 Random Search Results

Confusion Matrix

	0	1	2	3	4	5
0	964	64	57	452	55	17
1	150	1213	7	122	5	9
2	60	4	1490	25	2	2
3	328	45	7	749	11	9
4	26	3	2	22	1556	2
5	12	3	2	14	0	1527
	0	1	2	3	4	5
	Predicted					

Actual

Figure 6.15 Confusion Matrix of SVM

Random Forest Model and Evaluation: In the Random Forest approach, we employ an ensemble learning technique known as Random Forest Classifier. The RandomForestClassifier class from sklearn.ensemble is used to create this classifier.

Initially, the classifier is trained on the TF-IDF transformed training data using the fit() function. Predictions are then generated on the test data using the predict() function. The accuracy score is calculated. To gain a more comprehensive understanding of the model's performance, a classification report is generated.

```
from sklearn.ensemble import RandomForestClassifier
rf_clf = RandomForestClassifier()
rf_clf.fit(X_train_tf, y_train)
```

```
▼ RandomForestClassifier
RandomForestClassifier()
```

```
rf_pred = rf_clf.predict(X_test_tf)
```

```
accuracy = accuracy_score(y_test, rf_pred)
print(accuracy)
```

```
0.8359582963620231
```

Figure 6.16 Random Forest Implementation

Hyperparameter Tuning using RandomizedSearchCV

For enhancing the Random Forest model's performance, hyperparameter tuning is performed using RandomizedSearchCV. A set of hyperparameters, such as n_estimators, max_depth, min_samples_split, min_samples_leaf, bootstrap, and criterion, are defined for tuning. The search is performed with cross-validation using 5-fold cross-validation. The RandomizedSearchCV algorithm iterates over a predefined number of combinations to find the optimal hyperparameters.

After tuning, the model with the best parameters is identified and predictions are made on the test data using this optimized model.

```
# Define the parameter grid for RandomizedSearchCV
param_dist = {
    'n_estimators': [5,20,50,100],
    'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],
    'min_samples_split': [2, 6, 10],
    'min_samples_leaf': [1, 3, 4],
    'bootstrap': [True, False],
    'criterion': ['gini', 'entropy']
}

rf_clf = RandomForestClassifier()

random_search = RandomizedSearchCV(estimator=rf_clf,param_distributions=param_dist,
                                   n_iter=10,cv=5,scoring='accuracy',n_jobs=-1)

random_search.fit(X_train_tf, y_train)
```

RandomizedSearchCV

estimator: RandomForestClassifier

RandomForestClassifier

Figure 6.17 Random forest tuning

		Confusion Matrix					
Actual	0	1313	50	69	56	97	24
	1	294	1172	7	24	2	7
	2	53	3	1522	2	0	3
	3	800	55	38	162	82	12
	4	18	1	1	2	1587	2
	5	22	4	4	2	3	1523
		0	1	2	3	4	5
		Predicted					

Figure 6.18 Confusion Matrix of Random Forest

Naive Bayes Model and Evaluation: The Multinomial Naive Bayes classifier is chosen for this task. It's a probabilistic classification algorithm particularly suitable for text data. The training data, represented as TF-IDF vectors, is utilized to train the Naive Bayes classifier using the `fit()` function. Subsequently, predictions are made on the test data using the `predict()` function. The accuracy of the model is evaluated by comparing the predicted labels with the actual labels and calculating the accuracy score.

```
from sklearn.naive_bayes import MultinomialNB
nb_clf = MultinomialNB()
nb_clf.fit(X_train_tf, y_train)
```

```
▼ MultinomialNB
MultinomialNB()
```

```
nb_pred = nb_clf.predict(X_test_tf)
```

```
accuracy = accuracy_score(y_test, nb_pred)
print(accuracy)
```

```
0.7264862466725821
```

Figure 6.19 Naive Bayes Implementation

6.4 RESULTS AND COMPARISON

In this section, we present a comprehensive overview of the results obtained from our cyberbullying tweet detection project. We detail the performance of various machine learning algorithms, their accuracy after hyperparameter tuning, and the impact of preprocessing steps. Our evaluation process includes popular classifiers such as Support Vector Machine (SVM), Logistic Regression, Random Forest, and Naive Bayes. The results

provide insights into the effectiveness of each algorithm in accurately identifying cyberbullying content in tweets.

1. Logistic Regression:

- Initial Accuracy: 0.8027
- Accuracy after Removing Duplicates: 0.8304
- Accuracy after Hyperparameter Tuning (RandomizedSearchCV): 0.8333
- Accuracy after Hyperparameter Tuning (GridSearchCV): 0.8333

2. Support Vector Machine

- Initial Accuracy: 0.8313
- Accuracy after Removing Duplicates: 0.8266
- Accuracy after Hyperparameter Tuning (Random Search): 0.8317

3. Random Forest:

- Accuracy after Removing Duplicates: 0.8359
- Accuracy after Hyperparameter Tuning (Random Search): 0.8073

4. Naive Bayes:

- Accuracy: 0.7213

Benchmark Comparison:

We conducted a thorough comparison of our project's results against existing benchmark models to evaluate our performance and contributions to the field of cyberbullying detection. The benchmark models and their accuracies are as follows:

1. Benchmark Model 1 (Author: Jatan Varma):

- Logistic Regression (LR): 82.28%

CYBERBULLYING TWEET DETECTION

Data Science Academia – Sai Chamundeeswari Academy (c) Copyright (August 15th, 2023)

- Support Vector Machine (SVM): 82.20%

2. Benchmark Model 2 (Author: Phonphrm Thawatdamrongkit):

- Random Forest (RF) with CountVectorizer: 79.42%
- Random Forest (RF) with TF-IDF Vectorizer: 80.74%

3. Benchmark Model 3 (Author: Jhon Vincent Gupo):

- Naive Bayes (NB) after Oversampling: 85%
- Random Forest (RF) after Oversampling: 94%

Comparison Analysis:

- Our Support Vector Machine (SVM) achieved competitive accuracy, outperforming the benchmark SVM accuracy of 82.20%.
- Logistic Regression demonstrated improved accuracy after hyperparameter tuning, surpassing the benchmark accuracy of 82.28% for LR.
- Random Forest showcased remarkable accuracy after removing duplicates, exceeding the benchmark RF accuracies of 79.42% and 80.74%.
- Naive Bayes, while exhibiting a lower accuracy initially, remains an area for improvement.

7. CONCLUSION

Our project's results highlight the potential of machine learning techniques in identifying cyberbullying content within tweets. Our findings emphasize the critical role of hyperparameter tuning in elevating model accuracy. Through systematic tuning of parameters, we were able to extract optimal configurations for our algorithms, highlighting the significance of fine-tuning in achieving peak performance.

The versatility of Logistic Regression was showcased through its performance in both binary and multi-class classification scenarios. After meticulous tuning, this classifier achieved competitive accuracy, demonstrating its potential as a reliable choice for text classification tasks. Support Vector Machine (SVM), known for its resilience and versatility, demonstrated stability in performance after hyperparameter tuning. Its ability to handle non-linear relationships within data and its consistent accuracy highlight its value in cyberbullying detection.

Random Forest exhibited strong performance even before hyperparameter tuning. Its accuracy further improved after the removal of duplicate entries from the dataset, showcasing the importance of data preprocessing in enhancing model effectiveness. Naive Bayes, while demonstrating a relatively lower accuracy in our project, emphasizes the necessity of considering algorithm suitability for specific tasks. The performance differential between Naive Bayes and other algorithms highlights the importance of aligning algorithmic choices with the nature of the data and the intricacies of the problem being addressed. In conclusion, our endeavor to create a cyberbullying tweet detection model reveals significant advancements in the application of machine learning to the domain of online safety.

8. FUTURE WORK

As we continue our journey, we recognize the need for ongoing research and development to create even more robust cyberbullying detection models. Further exploration of ensemble methods, deep learning, and advanced preprocessing techniques will enable us to create safer online spaces by accurately identifying and combating cyberbullying. Adapting the models to detect cyberbullying in different languages can contribute to a more inclusive approach. This involves creating or leveraging multilingual datasets and considering linguistic nuances. One more future work would be creating user-friendly interfaces for reporting and analyzing cyberbullying which can encourage more users to engage and contribute to the safety of online spaces.

9. REFERENCES

1. Muneer, Amgad, and Suliman Mohamed Fati. "A comparative analysis of machine learning techniques for cyberbullying detection on twitter." *Future Internet* 12.11 (2020): 187.
2. Purnamasari, N. M. G. D., et al. "Cyberbullying identification in twitter using support vector machine and information gain based feature selection." *Indonesian Journal of Electrical Engineering and Computer Science* 18.3 (2020): 1494-1500.
3. Karan, Suman. Cyberbully: Aggressive Tweets, Bully and Bully Target Identification from Multilingual Indian Tweets. Diss. Indian Institute of Technology Jodhpur, 2021.
4. Waseem, Zeerak, and Dirk Hovy. "Hateful symbols or hateful people? predictive features for hate speech detection on twitter." Proceedings of the NAACL student research workshop. 2016.
5. Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the international AAAI conference on web and social media. Vol. 11. No. 1. 2017.
6. <https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification/code>
7. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
8. <https://www.mdpi.com/1999-5903/12/11/187>
9. <https://ijeecs.iaescore.com/index.php/IJECS/article/view/15105>
10. https://cse.iitj.ac.in/images/pdf/project-reports/MT19CS019_Thesis_Final.pdf
11. <https://github.com/t-davidson/hate-speech-and-offensive-language>
12. <https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/>
13. <https://www.kaggle.com/code/jonaspptawat/cyberbullying-classification-eda-and-ml#MACHINE-LEARNING>

14. <https://www.kaggle.com/code/jayantverma9380/cyberbullying-tweet-recognition-project>
15. <https://www.youtube.com/watch?v=jbexvUovHxw>
16. <https://github.com/t-davidson/hate-speech-and-offensive-language>
17. <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>
18. [https://machinelearningmastery.com/classification-as-conditional-probability-and-the-naive-bayes-algorithm/#:~:text=The%20conditional%20probability%20can%20be,A\)%20%2F%20P\(B\)](https://machinelearningmastery.com/classification-as-conditional-probability-and-the-naive-bayes-algorithm/#:~:text=The%20conditional%20probability%20can%20be,A)%20%2F%20P(B))
19. <https://scikit-learn.org/stable/modules/svm.html>