RACISM DETECTION ON TWITTER USING SENTIMENTAL ANALYSIS

A PROJECT REPORT

Submitted by

MOHANAPRIYA V	**510619104050**
PRIYADHARSHINI S	**510619104054**
SIVA PRIYA T	**510619104065**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ENGINEERING

C. ABDUL HAKEEM COLLEGE OF ENGINEERING AND TECHNOLOGY,

MELVISHARAM, VELLORE -632509

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report **"RACISM DETECTION ON TWITTER USING SENTIMENTAL ANALYSIS"** is the bonafide work of **"MOHANAPRIYA V (510619104050), PRIYADHARSHINI S (510619104054), SIVAPRIYA T (510619104065)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. K. ABRAR AHMED M.E, Ph.D.,	**Mrs. P. REVATHI M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Computer	Department of Computer
Science and Engineering	Science and Engineering
C. Abdul Hakeem College of	C. Abdul Hakeem College of
Engineering and Technology	Engineering and Technology
Melvisharam -632 509	Melvisharam -632 509

Submitted for the University Viva-Voce Examination held on ___________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At the outset, we would like to express our gratitude to our beloved and respond almighty, for his support and blessings to accomplish the project.

We would like to express our thanks to our Honorable Chairman **HAJI S. ZIAUDDIN AHMED SAHEB B.A.**, and to our beloved Correspondent **HAJI V.M. ABDUL LATHEEF SAHEB B.E, FIE.**, for his encouragement and guidance.

We thank our principal, **Dr. M. SADIQUE BASHA M.Tech, Ph.D.,** for creating a wonderful environment for us and enabling us to complete the project.

We wish to express our sincere thanks and gratitude to **Dr. K. ABRAR AHMED MTech., Ph.D.,** Head of the Department of Computer Science and Engineering who has been a guiding force and constant source of inspiration to us.

We wish to express our sincere thanks and gratitude to our beloved Project Guide **Mrs. P. REVATHI M.E.,** Assistant Professor, Department of Computer Science for having extended her fullest co-operation and guidance without which this project would not have been a success.

Our thanks to all faculty and non-teaching staff members of our department for their constant support to complete this project.

ABSTRACT

The Internet's widespread availability has drastically altered how we view the world. One of the most significant data sources for academics is Twitter, which is presently one of the top platforms among the several extant social networks. Social media may be utilized as real-world sensors to gauge the pulse of cultures. However, the vast and unfiltered stream of social media posts today raises societal concerns, particularly when these posts contain racism directed at a particular person or group. Social media, particularly Twitter, has been utilized in recent years to propagate anti-Racist messages. Governments and non-governmental organizations (NGOs) are concerned about the potential adverse effects that these messages may have on people or on society under this situation. In this study, we suggest Stanford NLP's Sentiment Analysis of Tweets for Racism Detection. The project uses Stanford NLP sentiment analysis to look for racism in tweets. The initial input is the Twitter dataset. Once the text data has been preprocessed, sentiment analysis is used to categorize tweets as racist or not racist. Through automatic moderation, the findings will be utilized to increase awareness and stop the propagation of prejudice online. Our key contribution is the use of Stanford NLP to obtain promising outcomes in the area of racism.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABLES	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATION	ix
1.	**INTRODUCTION**	1
	1.1 Overview of the project	1
	1.2 Purpose of the project	2
	1.3 Scope of the project	2
2.	**LITERATURE SURVEY**	3
	2.1 Historical Background	3
	2.2 Existing System	11
	2.3 Problem Statement	13
3.	**IDEATION AND PROPOSED SYSTEM**	14
	3.1 Empathy Map	14
	3.2 Ideation and Brainstorming	15
	3.3 Proposed System	15
	3.4 Problem Solution Fit	18
4.	**MODULES**	19
	4.1 Modules Involved	19
	4.2 Algorithm implementation	19
5.	**PROJECT DESIGN**	24
	5.1 UML DIAGRAMS	24
	5.1.1 Use Case Diagram	26

	5.1.2 Class Diagram	27
	5.1.3 Data Flow Diagram	25
	5.1.4 Activity Diagram	28
	5.1.5 Sequence Diagram	29
	5.2 SOLUTION & TECHNICAL ARCHITECTURE	30
	5.3 USER STORIES	30
6.	**SYSTEM REQUIREMENTS**	33
	6.1 Hardware Requirements	33
	6.2 Software Requirements	33
7.	**CODING AND SOLUTION**	36
8.	**PROJECT PLANNING AND SCHEDULING**	40
	8.1 Sprint Planning & Estimation	40
	8.2 Sprint Delivery Schedule	40
	8.3 Reports From JIRA	41
9.	**TESTING**	42
	9.1 Types of Testing	42
	9.2 User Acceptance Testing	45
10.	**RESULTS**	46
	10.1 Performance Metrics	46
11.	**CONCLUSION**	49
12.	**FUTURE SCOPE**	50
13.	**APPENDICES**	53
	Appendix-I- sample code	53
	Appendix-II – screenshots	66
	REFERENCES	72

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
5.1	User Stories	30
7.1	Pre-processing completion	37
8.2	Sprint Planning & estimation	40
8.3	Sprint Delivery Schedule	40

LIST OF FIGURES

FIG.NO	TITLE	PAGE NO.
3.1	Empathy Map	14
3.2	Brainstorming	15
3.3	Problem Solution Fit	18
5.1	Dataflow Diagram	24
5.2	Use Case Diagram	24
5.3	Class Diagram	26
5.4	Activity Diagram	27
5.5	Sequence Diagram	28
5.6	System Architecture	29
7.1	Sample text from dataset	36
7.2	Sentiment analysis	39
7.3	Static Graph	39
8.1	Report from JIRA	41
11.1	Admin Login page	66
11.2	Home page	66
11.3	Upload dataset	67
11.4	Data Uploaded	67
11.5	Data cleaning	68
11.6	Symbols Removal	68
11.7	Preprocessing	69
11.8	Stop words removal	69
11.11	Racism detection Analysis	71
11.12	Pictorial Representation	71

LIST OF ABBREVIATION

GRU	Gated Recurrent Unit
CNN	Convolution Neural Network
GCR-NN	Gated Convolution Recurrent Neural Network
RNN	Recurrent Neural Network
NLP	Natural Language Processing
BoW	Bag Of Words
UML	Unified Modeling Language
DFD	Data Flow Diagram
SQL	Structured Query Language
API	Application Programing Interface
IDE	Integrated Development Environment
GB	Gigabyte

1.INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The wide use of social media is a potential source of data generation containing important information regarding people's attitudes, responses, emotions, and opinions regarding specific events, objects, personalities, and entities. Social media sites, such as Twitter, represent a new setting in which racism and related stress are apparently prospering.

Racism detection on Twitter using Natural Language Processing (NLP) involves the use of machine learning algorithms to identify and classify racist content on the social media platform. The process typically involves several steps, including data collection, pre-processing, feature extraction, and model training and evaluation. Data collection involves gathering a large dataset of tweets from Twitter, either through public APIs or through web scraping. The collected tweets are then pre-processed to remove noise, such as stop words, punctuation, and URLs. The next step is to extract relevant features from the pre-processed data, such as n-grams, word embeddings, and sentiment analysis. The next step is to train a machine learning model on the extracted features to classify tweets as either racist or non-racist. The most commonly used models include logistic regression, Naive Bayes, and Support Vector Machines (SVMs). Once the model is trained, it is evaluated on a test dataset to measure its performance metrics such as accuracy, precision, recall, and F1-score.

1.2 PURPOSE OF THE PROJECT

The purpose of racism detection on social media platforms like Twitter is to identify and categorize tweets containing racist content using natural language processing techniques. The goal is to develop automated approaches to detect and remove harmful content from social media platforms to promote a more inclusive and respectful online community.

By detecting and removing racist content, social media platforms can create a safe and welcoming environment for people of all races and cultures. Racism detection can also help to identify patterns of racist behaviour and hate speech, which can be used to develop targeted interventions to prevent and combat racism. Additionally, the data generated from racism detection algorithms can provide insights into the prevalence and nature of racist behaviour on social media platforms, which can be used to inform policy and decision-making.

1.3 SCOPE OF THE PROJECT:

- To develop a model that can identify, categorize and detect tweets containing racist content.
- To analyze the context and meaning of texts, the tone and sentiment of language, and cultural references and slang that may be used to express racist sentiments.
- To provide valuable insights into the prevalence and nature of racist behavior, which can be used to inform policy and decision-making.
- The scope of racism detection on Twitter using NLP is focused on identifying and removing harmful content from social media platforms to promote a more inclusive and respectful online community.

2.LITERATURE SURVEY

2.1 HISTORICAL BACKGROUND

Survey 1:

Title: Using social media to understand and guide the treatment of racist ideology

Authors: K. R. Kaiser, D. M. Kaiser, R. M. Kaiser, and A. M. Rackham

Social media, including sites such as Face Book, Twitter and Instagram, provides a platform for racist ideology, making this dysfunction of American society more evident. Social media can provide insight into the world of the racist – individuals who cling to their tribal identities, irrationally rejecting those who they perceive as different. Studying social media may provide insight into processes that can assist in healing American society of its segregationist views – a way toward healing the racist. The purpose of this paper is to analyze social media posts to better understand racism, its causality, and to develop initial steps for addressing racist ideology. A qualitative review consisting of content analysis of 600 American Face Book posts was completed to reveal patterns in cognition, problem solving, personality structures, belief systems, and coping styles. The content analysis consists of both a descriptive account of the data and an interpretive analysis. Keywords: Racism, social media, violence, social conditioning, sexism, ageism, anti-Semitism, able-bodism, heterosexism, paranoia, Christianity, Cluster B Personality Traits, clandestine.

Survey 2:

Title: Using social media for health research: Methodological and ethical considerations for recruitment and intervention delivery

Authors: D. Arigo, S. Pagoto, L. Carter-Harris, S. E. Lillie, and C. Nebeker

As the popularity and diversity of social media platforms increases so does their utility for health research. Using social media for recruitment into clinical studies and/or delivering health behavior interventions may increase reach to a broader audience. However, evidence supporting the efficacy of these approaches is limited, and key questions remain with respect to optimal benchmarks, intervention development and methodology, participant engagement, informed consent, privacy, and data management.

Little methodological guidance is available to researchers interested in using social media for health research. In this Tutorial, we summarize the content of the 2017 Society for Behavioral Medicine Pre-Conference Course entitled 'Using Social Media for Research,' at which the authors presented their experiences with methodological and ethical issues relating to social media-enabled research recruitment and intervention delivery. We identify common pitfalls and provide recommendations for recruitment and intervention via social media. We also discuss the ethical and responsible conduct of research using social media for each of these purposes.

Survey 3:

Title: Online networks of racial hate: A systematic review of 10 years of research on cyber-racism

Authors: A.-M. Bliuc, N. Faulkner, A. Jakubowicz, and C. McGarty

The ways in which the Internet can facilitate the expression and spread of racist views and ideologies have been the subject of a growing body of research across disciplines. To date, however, there has been no systematic reviews of this research. To synthesize current knowledge on the topic and identify directions for future research, we systematically review a decade of research on cyber-racism as perpetrated by groups and individuals (i.e., according to the source of cyber-racism). Overall, the cyber-racism research reviewed shows that racist groups and individuals use different communication channels, are driven by different goals, adopt different strategies, and the effects of their communication are distinctive.

Despite these differences, both groups and individuals share a high level of skill and sophistication when expressing cyber-racism. Most of the studies reviewed relied on qualitative analyses of online textual data. Our review suggests there is a need for researchers to employ a broader array of methods, devote more attention to targets' perspectives, and extend their focus by exploring issues such as the roles of Internet in mobilizing isolated racist individuals and in enabling ideological clustering of supporters of racist ideologies.

Survey 4:

Title: Meta-analysis: Are psychotherapies less effective for black youth in communities with higher levels of anti-black racism

Authors: M. A. Price, J. R. Weisz, S. McKetta, N. L. Hollinsaid, M. R. Lattanner, A. E. Reid, and M. L. Hatzenbuehler

Objective: To examine whether anti-Black cultural racism moderates the efficiency of psychotherapy interventions among youth.

Method: A subset of studies from a previous meta-analysis of 5 decades of youth psychotherapy randomized controlled trials was analyzed. Studies were published in English between 1963 and 2017 and identified through a systematic search. The 194 studies (N = 14,081 participants; age range, 2-19) across 34 states comprised 2,678 effect sizes (ESs) measuring mental health problems (eg, depression) targeted by interventions.

Anti-Black cultural racism was operationalized using a composite index of 31 items measuring explicit racial attitudes (obtained from publicly available sources, eg, General Social Survey) aggregated to the state level and linked to the meta-analytic database. Analyses were conducted with samples of majority-Black (ie, ≥50% Black) (n = 36 studies) and majority-White (n = 158 studies) youth.

Survey 5:

Title: Reducing racial inequities in health: Using what we already know to take action

Authors: D. Williams and L. Cooper

This paper provides an overview of the scientific evidence pointing to critically needed steps to reduce racial inequities in health. First, it argues that communities of opportunity should be developed to minimize some of the adverse impacts of systemic racism. These are communities that provide early childhood development resources, implement policies to reduce childhood poverty, provide work and income support opportunities for adults, and ensure healthy housing and neighborhood conditions. Second, the healthcare system needs new emphases on ensuring access to high quality care for all, strengthening preventive health care approaches, addressing patients' social needs as part of healthcare delivery, and diversifying the healthcare work force to more closely reflect the demographic composition of the patient population. Finally, new research is needed to identify the optimal strategies to build political will and support to address social inequities in health. This will include initiatives to raise awareness levels of the pervasiveness of inequities in health, build empathy and support for addressing inequities, enhance the capacity of individuals and communities to actively participate in intervention efforts and implement large scale efforts to reduce racial prejudice, ideologies, and stereotypes in the larger culture that undergird policy preferences that initiate and sustain inequities.

Survey 6:

Title: Racial differences in weathering and its associations with psychosocial stress: The CARDIA study.

Authors: Sarah Forrestera, David Jacobsb, Rachel Zmorab, Pamela Schreinerb, Veronique Rogerc, Catarina I. Kiefea

Biological age (BA) is a construct that captures accelerated biological aging attributable to "wear and tear" from various exposures; we measured BA and weathering, defined as the difference between BA and chronological age, and their associations with race and psychosocial factors in a middle-aged bi-racial cohort. We used data from the Coronary Artery Risk in Young Adults study (CARDIA), conducted in 4 U.S. cities from 1985–2016 to examine weathering for adults aged 48–60 years. We estimated BA via the Klemera and Doubal method using selected biomarkers. We assessed overall and race-specific associations between weathering and psychosocial measures. For the 2694 participants included, Blacks had a BA (SD) that was 2.6 (11.8) years older than their chronological age while the average BA among Whites was 3.5 (10.0) years younger than their chronological age (Blacks weathered 6.1 years faster than Whites). Belonging to more social groups was associated with less weathering in Blacks but not Whites, and after multivariable adjustment, lower SES and more depressive symptoms were associated with more weathering among Blacks than among Whites. We confirmed racial differences in weathering, and newly documented that similar psychosocial factors may take a greater toll on the biological health of Blacks than Whites.

Survey 7:

Title: Twitter and Research: A Systematic Literature Review Through Text Mining

Authors: Amir Karami 1, Morgan Lundy 2, Frank Webb 3, and Yogesh k. Dwivedi

Researchers have collected Twitter data to study a wide range of topics. This growing body of literature, however, has not yet been reviewed systematically to synthesize Twitter-related papers. The Existing literature review papers have been limited by constraints of traditional methods to manually select and analyze samples of topically related papers. The goals of this retrospective study are to identify dominant topics of Twitter-based research, summarize the temporal trend of topics, and interpret the evolution of topics withing the last ten years. This study systematically mines a large number of Twitter-based studies to characterize the relevant literature by an efficient and effective approach. This study collected relevant papers from three databases and applied text mining and trend analysis to detect semantic patterns and explore the yearly development of research themes across a decade. We found 38 topics in more than 18,000 manuscripts published between 2006 and 2019. By quantifying temporal trends, this study found that while 23.7% of topics did not show a significant trend (P D => 0:05), 21% of topics had increasing trends and 55.3% of topics had decreasing trends that these hot and cold topics represent three categories: application, methodology, and technology. The contributions of this paper can be utilized in the growing field of Twitter-based research and are beneficial to researchers, educators, and publishers.

Survey 8:

Title: Is Racism a Fundamental Cause of Inequalities in Health?

Authors: **Jo C. Phelan and Bruce G. Link**

We previously proposed that socioeconomic status (SES) is a fundamental cause of health inequalities and, as such, that SES inequalities in health persist over time despite radical changes in the diseases, risks, and interventions that happen to produce them at any given time. Like SES, race in the United States has an enduring connection to health and mortality. Our goals here are to evaluate whether this connection endures because systemic racism is a fundamental cause of health inequalities and, in doing so, to review a wide range of empirical data regarding racial differences in health outcomes, health risks, and health-enhancing resources such as money, knowledge, power, prestige, freedom, and beneficial social connections.

We conclude that racial inequalities in health endure primarily because racism is a fundamental cause of racial differences in SES and because SES is a fundamental cause of health inequalities. In addition to these powerful connections, however, there is evidence that racism, largely via inequalities in power, prestige, freedom, neighborhood context, and health care, also has a fundamental association with health independent of SES.

2.2 EXISTING SYSTEM

The existing system study proposes an approach for racism detection on social media platforms using machine learning and deep learning technique. As the first step is crawled from Twitter, followed by data cleaning and preprocessing, and finally the data annotation. In the end, the proposed stacked ensemble model is trained and tested on the datasets and its performance is compared.

The existing system used Ensemble Model. An ensemble model is proposed that makes use of recurrent neural networks. For this purpose, gated recurrent unit (GRU), convolution neural network, and recurrent neural network are stacked to make the GCR-NN model to perform sentiment analysis.

The existing system experiments for sentiment analysis on racism tweets have been carried out using an Intel Corei7 11th generation machine operating on Windows 10. Machine learning and deep learning models are implemented on Jupyter in python language using Tensor-floow, Kara's, and Sci-kit learn frameworks.

2.2.1 DISADVANTAGES

Data Cleaning and Preprocessing Challenges:

The process of data cleaning and preprocessing, which involves removing irrelevant or noisy data, may be challenging when dealing with social media data. Social media posts are often short, informal, and contain user-generated content with variations in language, grammar, and spelling. This can result in difficulties in accurately cleaning and preprocessing the data, leading to potential biases or inaccuracies in the analysis.

Reliance on Annotation for Ground Truth:

The existing system mentions data annotation as one of the steps, but it does not elaborate on how the annotation is performed. Data annotation for racism detection can be subjective and challenging, as it requires human annotators to interpret the sentiment or intent behind social media posts. This subjectivity can introduce biases or inconsistencies in the labeled data, which can affect the accuracy and reliability of the trained models.

Performance of Ensemble Model:

While the existing system proposes an ensemble model (GCR-NN) for sentiment analysis, it does not provide detailed information about the performance of the model. Without thorough evaluation metrics and comparisons with other existing models or benchmarks, it is difficult to assess the effectiveness of the proposed ensemble model in detecting racism on social media platforms.

Hardware and Software Dependencies:

The existing system mentions that the experiments were conducted on an Intel Core i7 machine with specific operating system and software frameworks (TensorFlow, Keras, and scikit-learn). However, this can create dependencies on specific hardware and software, making it less scalable or adaptable to different environments or setups.

In conclusion, while the existing system proposes an approach for racism detection on social media using machine learning and deep learning techniques, it has limitations in terms of data source, data cleaning and preprocessing challenges, reliance on annotation for ground truth, performance evaluation of the ensemble model, and hardware/software dependencies. Addressing these limitations could improve the reliability and generalizability of the system for real-world applications.

2.2 PROBLEM STATEMENT DEFINITION

To develop an effective solution that can detect and address racism in various forms, including but not limited to verbal and written communication, behavior, and systemic biases, to promote a more inclusive and equitable society.

This solution must be able to accurately identify instances of racism, provide appropriate responses and interventions, and prevent future occurrences through education and awareness-raising.

It must also address the complexities and nuances of racism, including intersectionality, cultural differences, and historical contexts, and be sensitive to potential biases and ethical considerations.

3.IDEATION AND PROPOSED SYSTEM

3.1 EMPATHY MAP

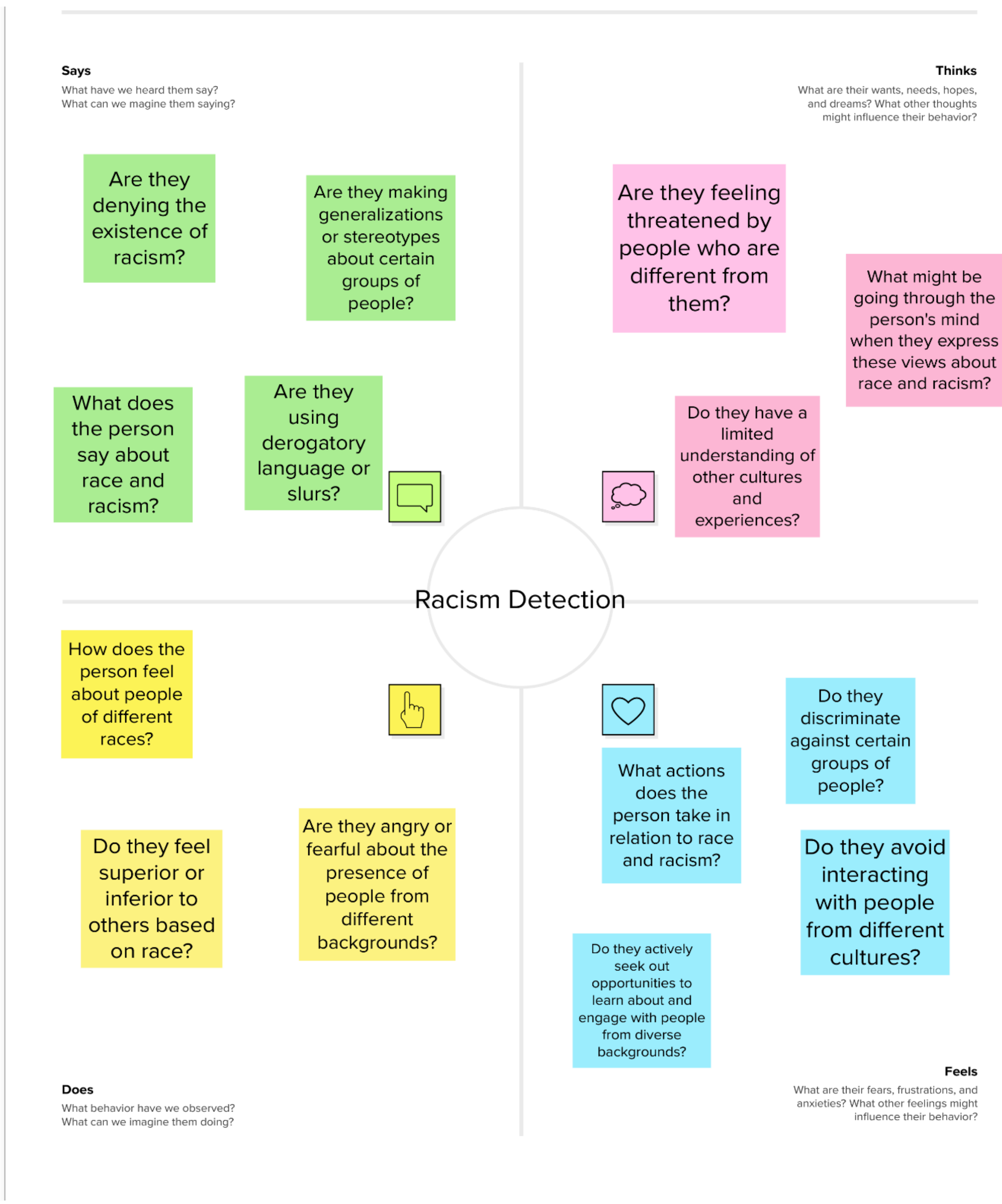


Fig 3.1: Empathy Map

3.2 IDEATION AND BRAINSTORMING

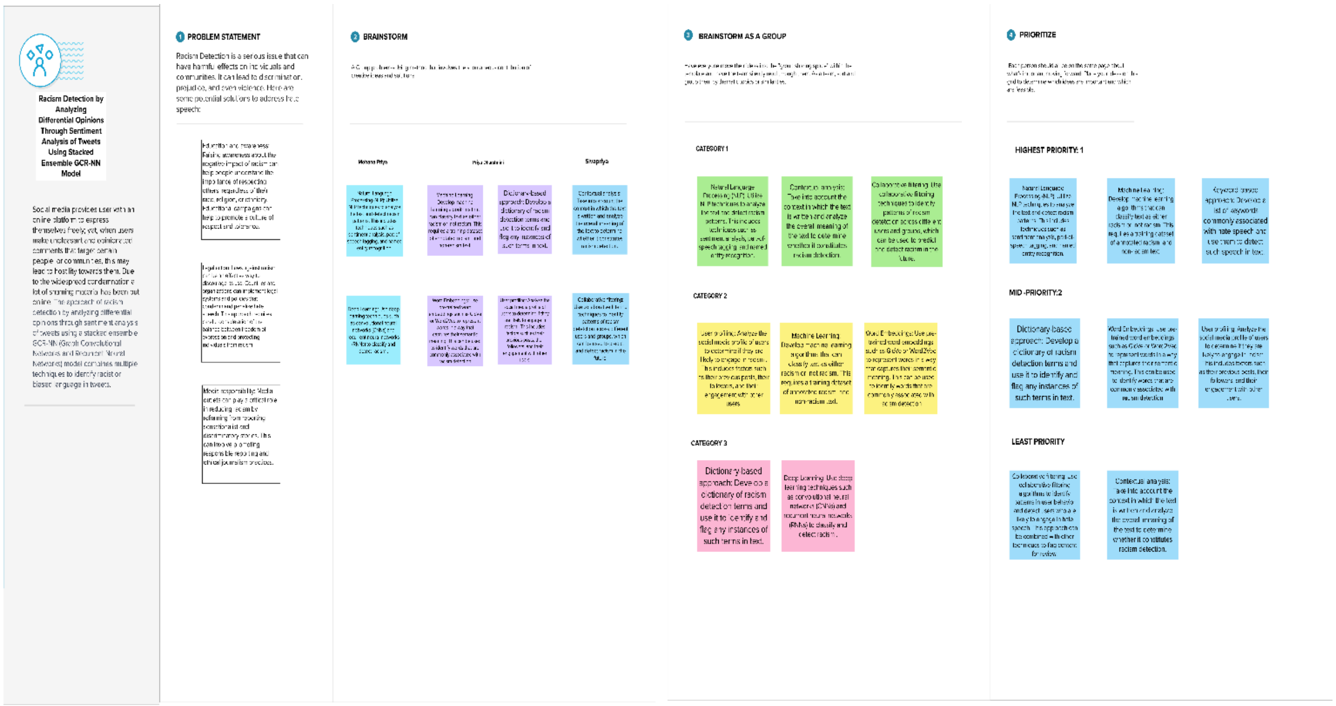


Fig 3.2: Brainstorming

3.3 PROPOSED SYSTEM

The project presents an approach for racism detection by analyzing differential opinions through sentiment analysis of tweets, developed using Java and NLP techniques. The proposed system involves several key steps. First, we collect the dataset from the Kaggle repository. The dataset doesn’t contain any label and it contains only the tweets posted in the twitter. Next, we perform data preprocessing to clean and prepare the tweets for analysis. We leverage NLP techniques to extract relevant features from the tweets, such as keywords, hashtags, and sentiment scores. We then apply a sentiment analysis algorithm to classify the tweets as positive, negative, or neutral based on their sentiment scores.

To implement our approach, we utilize Java programming language and various NLP libraries such as Stanford NLP and Apache Open NLP. We also incorporate natural language techniques, to improve the accuracy and performance of the system.

The proposed system has the potential to contribute to addressing the issue of racism in social media and beyond. The proposed system has the potential to be applied in various real-world scenarios, such as monitoring social media for racist content, identifying trends in public opinion on racism, and supporting efforts for combating racism.

3.3.1 ADVANTAGES

Accurate Racism Detection:

The proposed system utilizes NLP techniques and sentiment analysis algorithms to accurately detect racist content in tweets. By extracting relevant features such as keywords, and sentiment scores, the system can classify tweets as positive, negative, or neutral, enabling effective identification of racist content in social media.

Data Preprocessing for Improved Analysis:

The proposed system includes data preprocessing steps to clean and prepare the tweets for analysis. This ensures that the input data is properly processed and eliminates noise or irrelevant information, which can lead to more accurate results in the subsequent analysis.

Utilization of Java and NLP Libraries:

The system is developed using Java programming language and leverages popular NLP libraries such as Stanford NLP and Apache Open NLP. These libraries provide robust and efficient tools for natural language processing, enhancing the accuracy and performance of the system.

Potential for Real-world Applications:

The proposed system has the potential to be applied in various real-world scenarios, such as monitoring social media for racist content, identifying trends in public opinion on racism, and supporting efforts for combating racism. This can contribute to addressing the issue of racism in social media and beyond, making it a valuable tool for societal impact.

Flexibility in Dataset Selection:

The system collects the dataset from a publicly available repository, allowing for flexibility in choosing the dataset that best fits the specific needs of the project. This enables customization and adaptability to different contexts or domains, enhancing the system's applicability in various scenarios.

Integration of Natural Language Techniques:

The proposed system incorporates natural language techniques, such as sentiment analysis, which can provide deeper insights into the sentiment and emotions expressed in tweets related to racism. This can help in understanding the nuances and complexities of racist content, and provide valuable information for further analysis and decision-making.

In conclusion, the proposed system for racism detection through sentiment analysis of tweets, developed using Java and NLP techniques, offers several advantages including accurate detection, data preprocessing, utilization of Java and NLP libraries, potential for real-world applications, flexibility in dataset selection, and integration of natural language techniques. These advantages make the proposed system a valuable tool for addressing the issue of racism in social media and beyond.

3.4 PROBLEM SOLUTION FIT

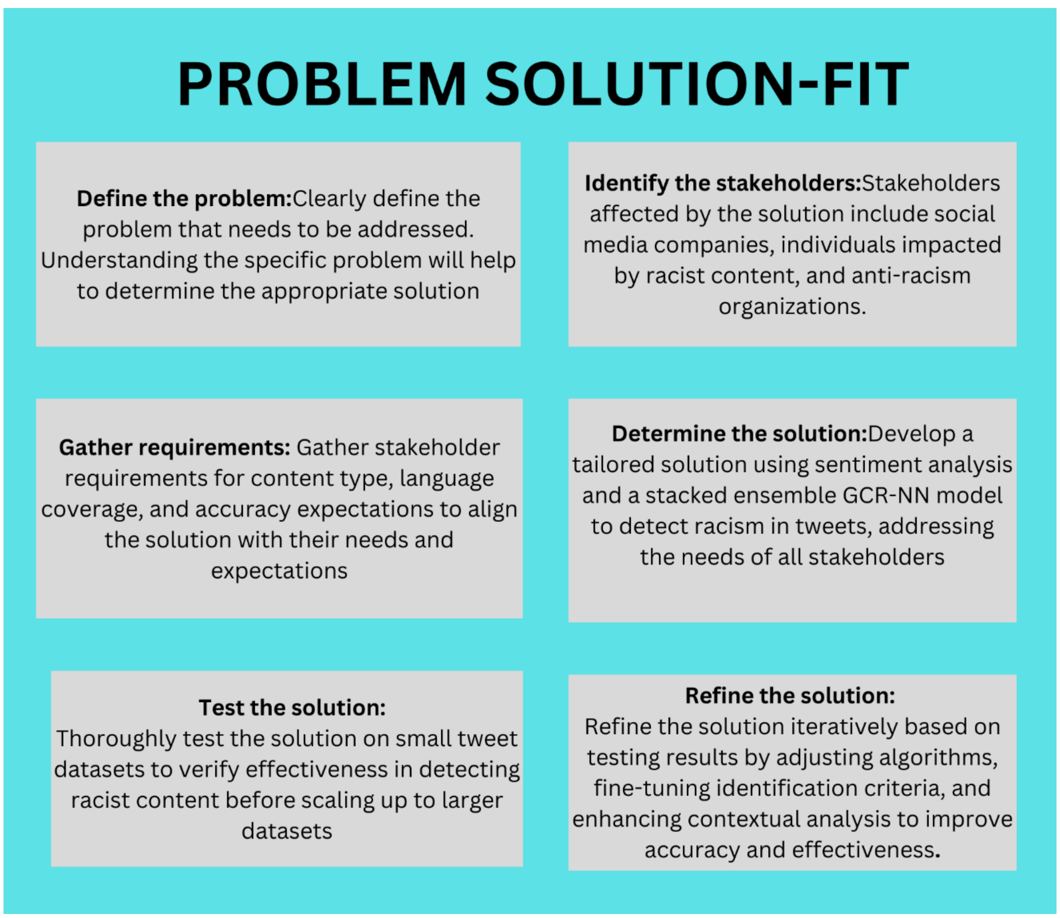


Fig 3.3: Problem Solution Fit

4. MODULES DESCRIPTION AND ALGORITHM

4.1 MODULES INVOLVED:

- Dataset Collection
- Input Dataset
- Data Tokenization
- Stop Words Exclusion
- BoW
- Sentiment Analysis
- Static Graph module

4.1.1 MODULES DESCRIPTION:

Dataset Collection:

The racism tweets dataset is collected from Twitter. Twitter has been the first choice of most researchers for text and sentiment analysis due to its being the most common platform widely used by a large number of people to express their feelings, views, comments, and opinions. This study intends to study the racism trends based on Twitter posts. For data collection, tweets related to racist comments have been collected. This module involves collecting the dataset from a publicly available repository, such as Kaggle, which contains tweets posted on Twitter. The collected data may not have any labels and would require further processing for analysis.

Input Dataset:

In this module we develop our proposed system to accept the input dataset which is referred from Kaggle. This dataset doesn't contain any label. It takes the tweets as input and uses natural language processing techniques to identify the racism tweets by analyzing it.

Data Tokenization:

This module focuses on cleaning and preparing the collected data for analysis. It may involve tasks such as removing irrelevant information, handling missing data, and normalizing the text data. Data preprocessing is crucial for ensuring the accuracy and reliability of the subsequent analysis. In this module, we make the Data Tokenization process. Tokenization is the process of splitting natural texts into tokens without any white spaces. It involves breaking sentences down into constituent words set. Although looks simpler and straightforward, deciding which tokens are appropriate is not a trivial task.

Stop Words Exclusion:

In this module we perform the Stop Words Exclusion process. Stop words are words that do not contribute to the training of the system. Instead, they create complexity by increasing the feature space. So, stop words remove common words that are unlikely to contribute to the meaning of the text, such as a, am, an, the, and is, etc., Stop word removal helps reduce noise in the data and make it more efficient to analyze.

BoW:

The BoW is another commonly used feature extraction used in NLP tasks. It is the most convenient and adaptable approach to get a document's features. The Word's histogram within the text is examined in BoW. The frequency of the words is employed as a function for the training of the set. The BoW (Bag of Words) approach is implemented in this study by utilizing the Count Vectorizer. The technique of obtaining numerical vectors by transforming a textual data set is termed vectorization. The frequency of words is counted indicating that tokens have been counted and making the token vectors. The BoW assigns a value to every attribute based on the frequency of those features.

Sentiment Analysis:

This module involves applying a sentiment analysis NLP (Natural Language Processing) algorithm to classify the tweets as positive, negative, or neutral based on their sentiment scores. It uses Natural Language Processing Stanford model to accurately determine the sentiment expressed in the tweets related to racism.

Static Graph Module:

Static Graph Module is responsible for plotting static graphs with the results obtained from the analysis. It generates a visual representation of the total tweets and racism tweets identified from it. Static graphs are useful when the data being visualized is not changing frequently or when a single snapshot of the data is sufficient for the analysis. They can be saved as image files and included in reports, presentations, or publications, or used in web pages or other digital media.

4.2 ALGORITHM IMPLEMENTATION

Natural Language Processing:

NLP (Natural Language Processing) is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and human language. NLP techniques use computational methods to analyse, model, and understand natural language text or speech.

NLP techniques can be used for a variety of tasks, including language translation, sentiment analysis, text summarization, speech recognition, and language generation. NLP algorithms typically involve several steps, including text pre-processing, feature extraction, and modeling. NLP techniques are powerful tools for analyzing and understanding human language. They are widely used in a variety of applications, from chatbots and virtual assistants to social media analysis and medical text mining.

Naive Bayes Classifier:

This is a probabilistic classifier that predicts the probability of a particular label or category given the input data. It is often used for text classification tasks such as spam detection, sentiment analysis, and topic classification.

Classification Phase:

Feature Vector Creation: For a new, unlabeled instance (e.g., a tweet), the same feature extraction process is applied to create a feature vector.

Posterior Probability Calculation: Using Bayes' theorem, the algorithm calculates the posterior probability of each sentiment class given the observed features. It combines the prior probabilities and the conditional probabilities of the features for each class.

Class Assignment: The algorithm assigns the instance to the sentiment class with the highest posterior probability.

Support Vector Machines (SVM):

This is a classification algorithm that separates data points into different classes using a hyperplane in a high-dimensional space. It is often used for text classification tasks such as sentiment analysis and topic classification. SVM optimizes the placement of the decision boundary by solving an optimization problem. It seeks to minimize the classification error while maximizing the margin between classes. This optimization problem involves solving a quadratic programming task.

Once the SVM model is trained, it can classify new, unlabeled instances based on their feature vectors. The model assigns a sentiment class based on which side of the decision boundary the instance falls.

Static Graph Plotting:

Matplotlib:

This is a Python library for creating static, interactive, and animated visualizations in Python. It provides a wide range of graph types, including line, scatter, bar, and pie charts, as well as 3D plotting.

5. PROJECT DESIGN

5.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

5.1.1 DATA FLOW DIAGRAM

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details.

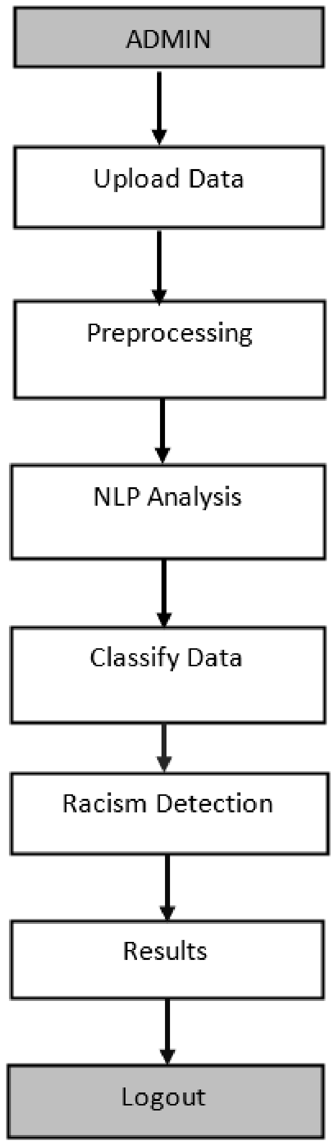


Fig 5.1: Dataflow Diagram

5.1.2 USE CASE DIAGRAM

A Use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

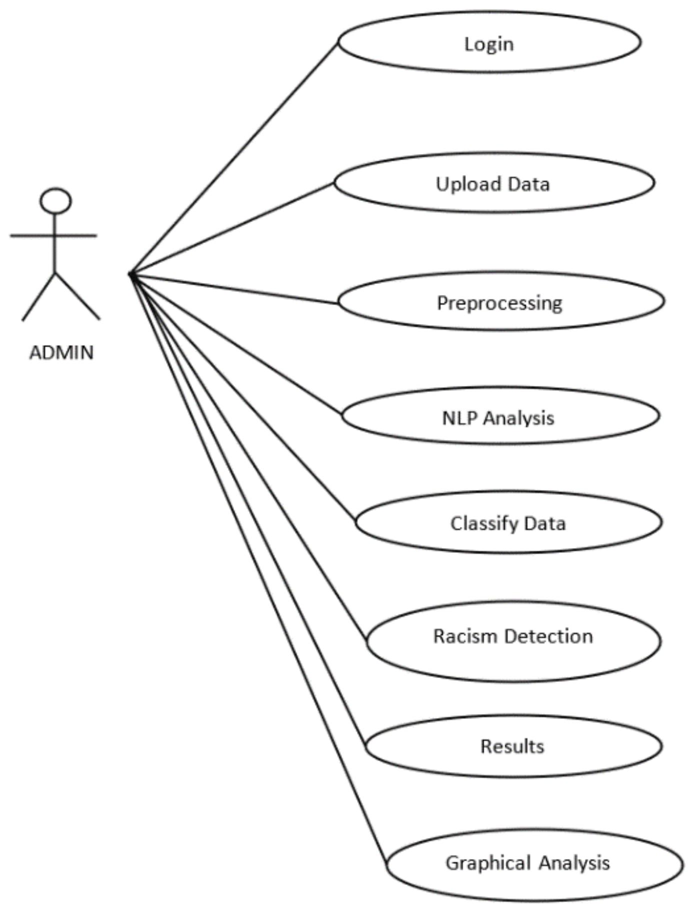


Fig 5.2: UseCase Diagram

5.1.3 CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

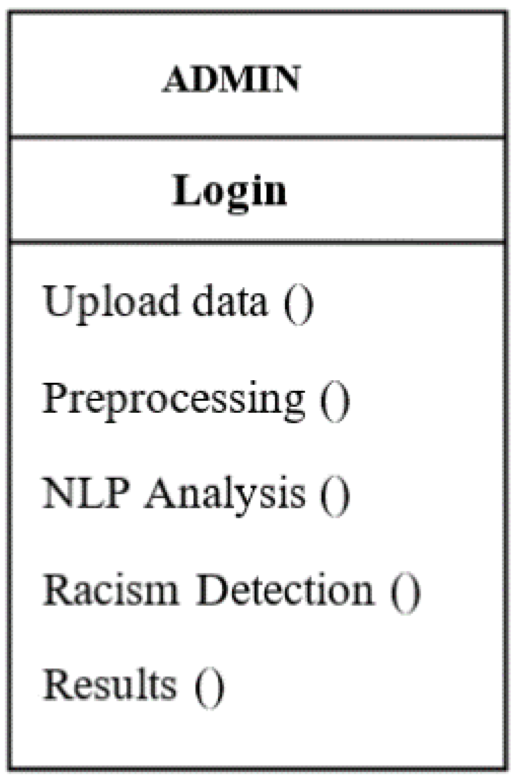


Fig 5.3: Class Diagram

5.1.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

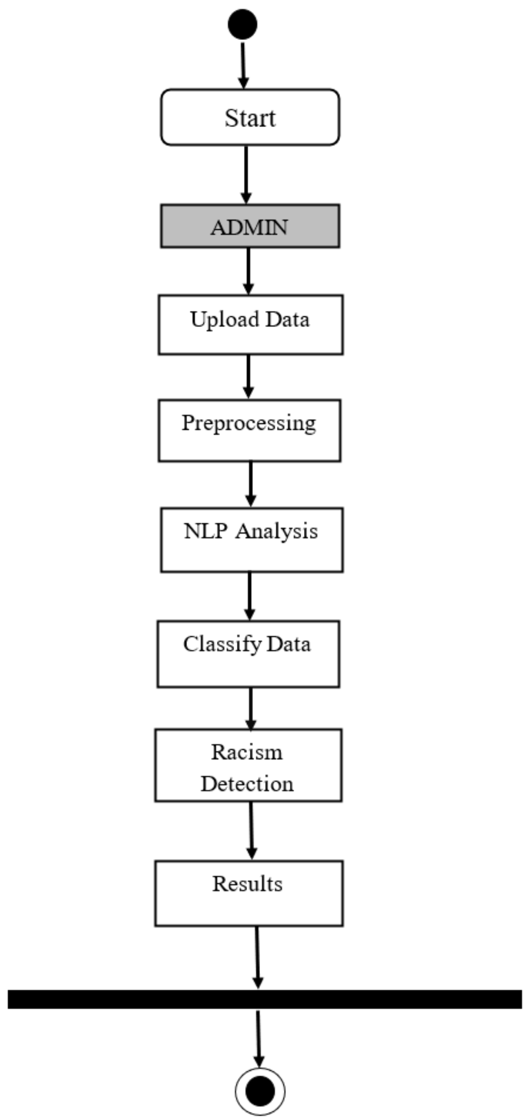


Fig 5.4: Activity Diagram

5.1.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

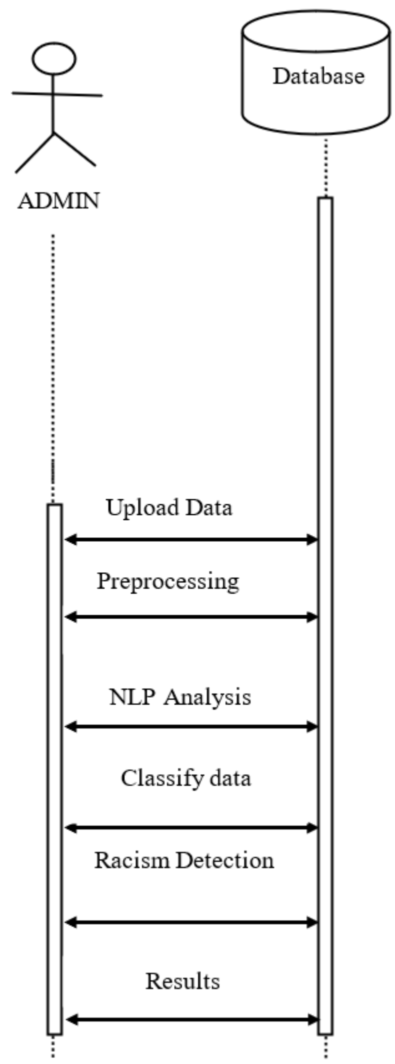


Fig 5.5: Sequence Diagram

5.2 SYSTEM ARCHITECTURE

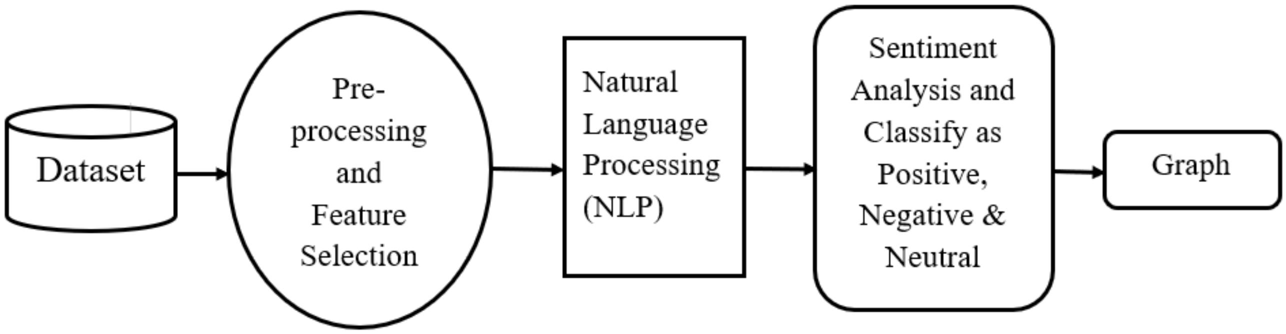


Fig 5.6: System Architecture

5.3 USER STORIES

USER TYPE	USER STORY NUMBER	USER STORY/ TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
ADMIN	USR-1	As a admin of the system, I want to upload the datasets which comprise of	The admin should be able to upload dataset by clicking "Upload" button and should receive pop-up message of successful attempt.	High	Sprint-1

		numerous tweets from the local disk to detect the racism content			
	USR-2	As a admin of the system, I want to enable the "Data Cleaning" process to the tweets.	The system will remove the symbols like “@, #, ", ', ;, :” for the purpose of simplification of tweets.	Medium	Sprint-2
	USR-3	As a Admin of the system, I want my	The system enables the Pre-Processing and the remove the stop words like	Medium	Sprint-3

		cleaned dataset to be pre-processed by clicking the "Pre-Processing" button in order to remove stop words.	'the', 'is', 'and' to eliminate unimportant words, allowing applications to focus on the important words instead.		
	USR-4	As a admin of system, I want to enable “analysis and detection” process to cleaned dataset.	The system will analyse and detect those cleaned dataset to classify them racism or non-racism.	High	Sprint-4

Table no.5.1 User story

6. SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS:

- System : Pentium i3 Processor
- Hard Disk : 500 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

6.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 10.
- Coding Language : JAVA.
- Tool : Apache NetBeans IDE 16
- Database : MYSQL

SOFTWARE ENVIRONMENT

JAVA PROGRAMING LANGUAGE :

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance

- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed.

THE JAVA PLATFORM

Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

JVM is the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

DATABASE:

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

MYSQL DATABASE:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

MySQL is released under an open-source license. So you have nothing to pay to use it.

MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.MySQL uses a standard form of the well-known SQL data language.

MySQL works on many operating systems and with many languages including HP, PERL, C, C++, JAVA, etc.MySQL works very quickly and works well even with large data sets.

MySQL is very friendly to PHP, the most appreciated language for web development.

7. CODING AND SOLUTIONING

PROPOSED METHODOLOGY

This study proposes an approach for racism detection on social media platforms using natural language processing. As the first step is crawled from Twitter, followed by data cleaning and preprocessing and finally the data annotation.

DATASET DESCRIPTION

The racism tweets dataset is collected from Twitter. For data collection, tweets related to racist comments have been collected. For this purpose, several keywords are used such as, `#racism', `#racial', and `#racist' etc.

@jncatron @isra_jourisra @AMPalestine Islamophobia is like the idea of Naziphobia. Islam is a religion of hate and it must be outlawed.
Finally I'm all caught up, and that sudden death cook off looks like it's gonna be intense #MKR
@carolinesinders @herecomesfran *hugs*
Humanity is still alive ! black man is forced into coffin by white south africans

Fig.no:7.1 Sample text from dataset

DATA PREPROCESSING:

Several steps are carried out at the preprocessing level to clean the data. It is vital to preprocess and clean the document adequately so a model can be trained appropriately.

Tokenization: This involves breaking the text into individual words or phrases, known as tokens. Tokenization helps standardize the text data and makes it easier to analyze.

Stop Word Removal: This involves removing common words that are unlikely to contribute to the meaning of the text, such as "the", "and", and "is". Stop word removal helps reduce noise in the data and make it more efficient to analyze.

Stemming: This involves reducing words to their root form, such as "running" to "run" or "jumps" to "jump". Stemming helps standardize the text data and reduces the number of unique words that need to be analyzed.

Lemmatization: This is similar to stemming but involves reducing words to their base form using a dictionary. For example, "am", "are", and "is" would all be reduced to "be". Lemmatization helps improve the accuracy of the analysis by reducing the number of variations of the same word.

Part-of-Speech (POS) Tagging: This involves labeling each word in a text data with its corresponding part of speech, such as noun, verb, or adjective. POS tagging can help identify patterns in the data and improve the accuracy of analysis.

Spell Checking: This involves correcting misspelled words in the text data. Spell checking helps improve the accuracy of the analysis by reducing noise in the data.

BEFORE PRE-PROCESSING	AFTER PRE-PROCESSING
@LeBale racism is good	racism good
@FBI it is clear to hundreds of millions of people of all walks that this country has a severe problem with systemic racism. your denial is discussing.	clear hundred million people walk country sever problem system racism denial

Table 7.1 Pre-Processing Completion.

FEATURE EXTRACTION:

BoW is used for features extraction to train the machine learning models. Features are extracted from the preprocessed tweets. One common approach is to represent each tweet as a Bag of Words, where each word is considered as a feature. The frequency of each word in the tweet is recorded, and this forms the feature vector for the tweet.

SENTIMENT ANALYSIS:

Training: The Naive Bayes classifier is then trained on the labeled training data. It calculates the prior probabilities of each sentiment class (positive, negative, neutral) and the likelihood probabilities of observing each feature (word) in each sentiment class. The classifier assumes that the features (words) are conditionally independent given the sentiment class, which is a naive assumption (hence the name "naive" Bayes).

Classification: Once the classifier is trained, it can be used to classify new, unlabeled tweets. The tweet is preprocessed in the same way as the training data, and the feature vector is constructed. The classifier calculates the posterior probability of each sentiment class given the observed features (using Bayes' theorem) and assigns the tweet to the sentiment class with the highest probability.

Sentiment Analysis: Based on the assigned sentiment class, the tweet is classified as positive, negative, or neutral. The sentiment scores can also be used to quantify the strength or intensity of the sentiment.

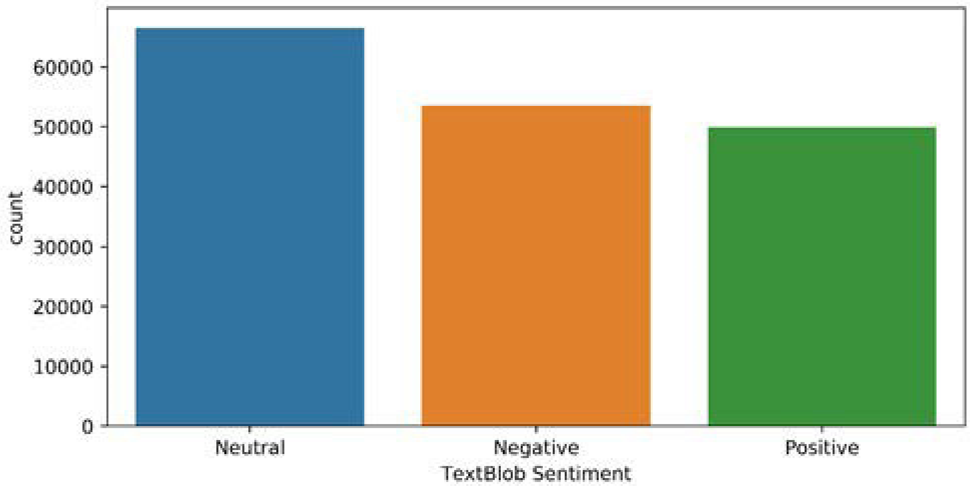


Fig.7.2: Sentiment Analysis

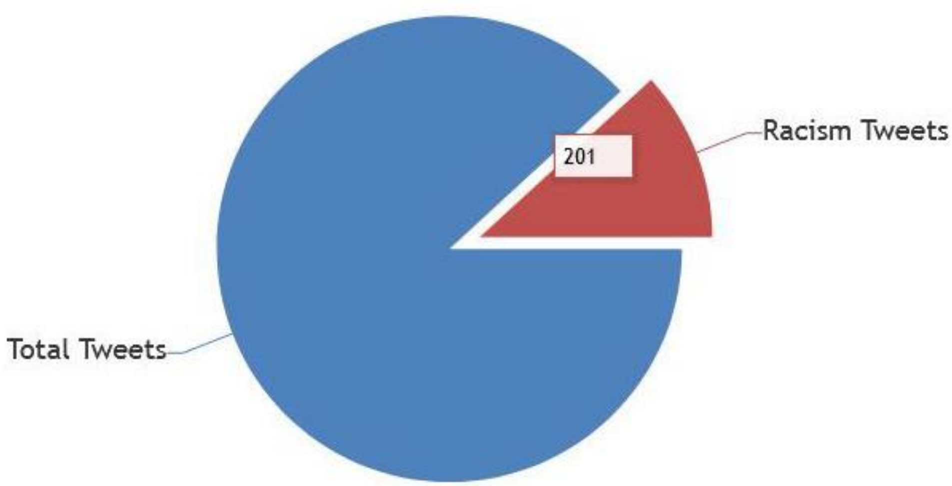


Fig 7.3: Static Graph

8.PROJECT PLANNING & SCHEDULING

8.1 SPRINT PLANNING & ESTIMATION

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Jan 2023	29 Jan 2023	20	29 Jan 2023
Sprint-2	20	6 Days	31 Jan 2023	05 Feb 2023	20	05 Feb 2023
Sprint-3	20	6 Days	07 Feb 2023	12 Feb 2023	20	12 Feb 2023
Sprint-4	20	6 Days	14 Feb 2023	19 Feb 2023	20	19 Feb 2023

Table no.8.1 Sprint Planning & Estimation

8.2 SPRINT DELIVERY SCHEDULE

SPRINT	USER STORY NUMBER	USER STORY/TASK	STORY POINT	PRIORITY	TEAM MEMBERS
Sprint-1	USN-1	Admin wants to upload the datasets which comprise of numerous tweets from the local disk to detect the racism content	2	High	Siva Priya T Mohana Priya V Priyadharshini S

Sprint-2	USN-2	Admin wants to enable the "Data Cleaning" process to the tweets.	2	Medium	Siva Priya T Mohana Priya V Priyadharshini S
Sprint-3	USN-3	Cleaned dataset to be pre-processed by clicking the "PreProcessing" button in order to remove stop words.	2	Medium	Siva Priya T Mohana Priya V Priyadharshini S
Sprint-4	USN-4	Admin wants to enable “analysis and detection” process to cleaned dataset.	1	High	Siva Priya T Mohana Priya V Priyadharshini S

Table No.8.2 Sprint Delivery Schedule

8.3 REPORT FROM JIRA

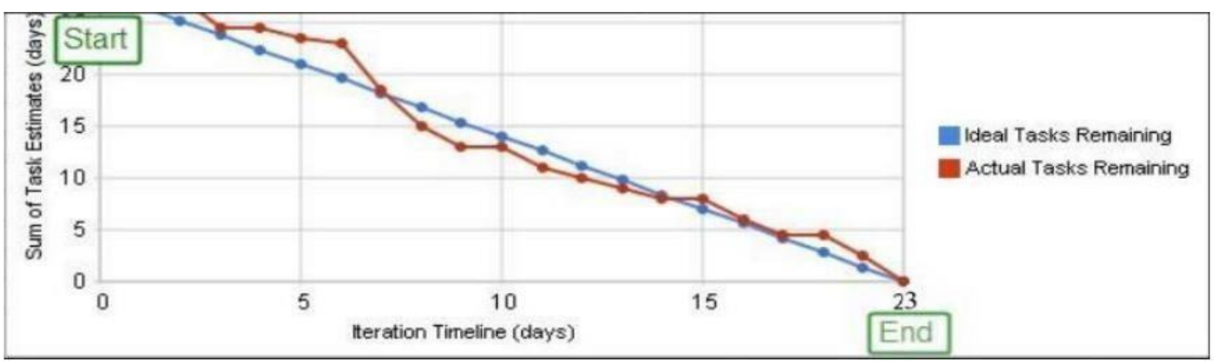


Fig No. 8.1 Report from JIRA

9.TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub- assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of testing. Each test type addresses a specific testing requirement.

9.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

9.2 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10. RESULT

10.1 PERFORMANCE METRICS

Precision and recall are important performance metrics for evaluating the effectiveness of racism detection systems. They provide insights into the system's ability to correctly identify instances of racism and avoid false positives and false negatives.

Precision measures the proportion of correctly identified racist instances out of all instances identified as racist.

It is calculated using the formula:

Precision = True Positives / (True Positives + False Positives)

Precision indicates how reliable the system is when it classifies an instance as racist.

A high precision score means that a large proportion of instances identified as racist are indeed racist, reducing the number of false positives.

Recall, also known as sensitivity or true positive rate, measures the proportion of correctly identified racist instances out of all actual racist instances in the data.

The formula for recall is:

Recall = True Positives / (True Positives + False Negatives)

Recall indicates the system's ability to identify and capture instances of racism from the dataset.

A high recall score means that the system can successfully identify a large proportion of actual racist instances, reducing the number of false negatives.

Precision and recall are complementary metrics, and their values need to be balanced depending on the specific goals and requirements of the racism detection system.

In some cases, precision may be more important, such as when the consequences of false positives are significant.

In other cases, recall may take precedence, particularly when it is crucial to capture as many instances of racism as possible, even at the cost of some false positives.

To get a more comprehensive evaluation of the system's performance, the F1 score is often used, which is the harmonic mean of precision and recall.

It provides a balanced measure that considers both metrics.

The F1 score is a widely used performance metric for racism detection (and other classification tasks) that balances both precision and recall.
It is particularly useful when there is an imbalance between the number of racist and non-racist instances in the dataset.

The F1 score is calculated using the formula:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

The F1 score ranges between 0 and 1, with 1 being the best possible score indicating perfect precision and recall.

It is the harmonic mean of precision and recall, giving equal weight to both metrics.

The F1 score is especially valuable when the racism detection system needs to balance minimizing false positives (non-racist instances classified as racist) and false negatives (racist instances classified as non-racist).

It provides an aggregate measure of performance that considers both types of errors.

By using the F1 score, practitioners can assess the overall effectiveness of their racism detection system, taking into account both precision and recall simultaneously.

This metric helps in finding the optimal balance between correctly identifying instances of racism and avoiding misclassifications.

Features	**Model**	**Accuracy**	**CP**	**WP**
BOW	DT	0.55	7,427	5,956
	RF	0.87	11,647	1,736
	KNN	0.28	3,868	9,515
	SVM	0.96	12,929	484
	LR	0.95	12,733	650

11. CONCLUSION

In conclusion, the proposed system presents an innovative approach for addressing the issue of racism by analyzing differential opinions through sentiment analysis of tweets. The project utilizes Java programming language and NLP techniques to collect, preprocess, and analyze tweet data from a publicly available repository. The system incorporates various NLP tasks such as feature extraction and sentiment analysis to classify tweets as positive, negative, or neutral based on their sentiment scores.

The proposed system has the potential to contribute to combating racism in social media and beyond. By automatically detecting racist content in tweets, it can help in monitoring social media for discriminatory language, identifying trends in public opinion on racism, and supporting efforts to raise awareness and take necessary actions to combat racism. Moreover, the system can be extended to other real-world applications such as identifying hate speech, bias, and discrimination in other forms of text data, and supporting decision-making in organizations working towards promoting diversity and inclusion.

Overall, the proposed project presents a valuable contribution to the field of NLP and social media analysis, with the potential to aid in addressing the issue of racism and promoting diversity and inclusiveness in our society.

12. FUTURE SCOPE

Here are some potential future directions for further work in the proposed project on racism detection through sentiment analysis of tweets using Java and NLP techniques:

Enhanced Feature Extraction:

The system can be further improved by exploring more advanced feature extraction techniques, such as word embeddings, topic modeling, or entity recognition, to capture the nuances of tweet content more accurately.

This may help in improving the accuracy of sentiment analysis and better identifying racist content.

Advanced Machine Learning Algorithms:

The project can be extended by exploring more advanced machine learning algorithms, such as deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or transformer-based models like BERT, to improve the accuracy and performance of the system.

These advanced algorithms may be able to capture complex patterns and dependencies in tweet data, leading to better racism detection.

Addressing Bias in Data and Models:

Bias is an important consideration in any NLP project, and efforts can be made to identify and mitigate potential biases in the dataset used for training and testing the system.

This may involve using techniques such as adversarial training, re-sampling, or re-balancing of data to ensure fair and unbiased results.

Additionally, bias detection and mitigation techniques can be integrated into the system to identify and mitigate bias in the sentiment analysis algorithm itself.

Real-time Monitoring and Alerting:

The system can be further extended to enable real-time monitoring of social media platforms for racist content.

This may involve integrating the system with APIs or tools that provide real-time access to tweet data, and implementing automated alerting mechanisms to notify relevant stakeholders when potentially racist tweets are detected.

This can aid in timely response and mitigation of racist content on social media.

Deployment and Evaluation in Real-world Settings:

Future work can involve deploying the proposed system in real-world settings to evaluate its effectiveness and impact.

This may involve collaborating with organizations, communities, or social media platforms to implement the system and collect feedback from users.

Rigorous evaluation can be conducted to assess the system's performance, accuracy, and usefulness in addressing the issue of racism in real-world scenarios.

User Interface and User Experience Improvements:

The usability and user experience of the system can be further enhanced by developing a user-friendly interface that allows users to interact with the system easily.

This may involve incorporating visualization techniques, providing informative feedback to users, and incorporating user feedback into the system's design and functionalities.

Overall, the proposed project has significant potential for further improvements and extensions, and future work can focus on addressing these areas to enhance the accuracy, robustness, and usability of the system in detecting racism in tweets and contributing to the efforts against racism in social media and beyond.

APPENDICES

APPENDIX 1 – SAMPLE CODE

Home page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- basic -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- mobile metas -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1">
    <!-- site metas -->
    <title>Analysis & Detection</title>
    <meta name="keywords" content="">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- bootstrap css -->
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
    <!-- style css -->
    <link rel="stylesheet" type="text/css" href="css/style.css">
    <!-- Responsive-->
    <link rel="stylesheet" href="css/responsive.css">
    <!-- fevicon -->
    <link rel="icon" href="images/fevicon.png" type="image/gif" />
```

```
        <!-- Scrollbar Custom CSS -->
        <link rel="stylesheet" href="css/jquery.mCustomScrollbar.min.css">
        <!-- Tweaks for older IEs-->
        <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-
awesome/4.0.3/css/font-awesome.css">
        <!-- fonts -->
        <link
href="https://fonts.googleapis.com/css?family=Poppins:400,700|Raleway:400,700
&display=swap" rel="stylesheet">
        <!-- owl stylesheets -->
        <link rel="stylesheet" href="css/owl.carousel.min.css">
        <link rel="stylesheet" href="css/owl.theme.default.min.css">
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.cs
s" media="screen">
     </head>
     <body>
        <!--header section start -->
        <div class="header_section">
           <div class="container-fluid">
              <nav class="navbar navbar-expand-lg navbar-light bg-light">
                 <div class="logo"><a href=""></a></div>
                 <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
```

```
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item" >
      <a class="nav-link" href="Home.jsp">Home</a>
    </li>
    <li class="nav-item" style="width: 200px ">
      <a class="nav-link" href="uploaddata.jsp">Upload Dataset</a>
    </li>
    <li class="nav-item" style="width: 200px ">
      <a class="nav-link" href="LoadedData.jsp">Data Cleaning</a>
    </li>
   <li class="nav-item" style="width: 200px ">
      <a class="nav-link" href="DataCleaning.jsp">Pre-Processing</a>
    </li>
    <li class="nav-item" style="width: 200px ">
      <a class="nav-link" href="Detection.jsp">Racism Detection</a>
    </li>
    <li class="nav-item" style="width: 100px ">
      <a class="nav-link" href="Graph.jsp">Graph</a>
    </li>
```

```html
                    <li class="nav-item" style="width: 100px ">
                        <a class="nav-link" href="index.jsp">Logout</a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>
</div>
<!--header section end -->
<!-- contact section start -->
<div class="contact_srction layout_padding">
    <div class="container">
        <h1 class="contact_taital">Racism Detection</h1>
        <div class="contact_srction_2">
            <div class="mail_main">
                <img src="images/detection.png">
            </div>
        </div>
    </div>
</div>
<!-- contact section end -->
<!-- footer section start -->
<div class="footer_section">
    <div class="container">
    </div>
</div>
```

```
<!-- footer section end -->
<!-- copyright section start -->
<div class="copyright_section">
   <div class="container">
      <p class="copyright_text">Copyright 2023 All Rights Reserved.<a
href="">Racism Detection</a></p>
   </div>
</div>
<!-- copyright section end -->
<!-- Javascript files-->
<script src="js/jquery.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
<script src="js/jquery-3.0.0.min.js"></script>
<script src="js/plugin.js"></script>
<!-- sidebar -->
<script src="js/jquery.mCustomScrollbar.concat.min.js"></script>
<script src="js/custom.js"></script>
<!-- javascript -->
<script src="js/owl.carousel.js"></script>
<script
src="https:cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.js">
</script>
</body></html>
```

Result page:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- basic -->
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <!-- mobile metas -->
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="viewport" content="initial-scale=1, maximum-scale=1">
        <!-- site metas -->
        <title>Analysis & Detection</title>
        <meta name="keywords" content="">
        <meta name="description" content="">
        <meta name="author" content="">
        <!-- bootstrap css -->
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
        <!-- style css -->
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <!-- Responsive-->
        <link rel="stylesheet" href="css/responsive.css">
                <link rel="stylesheet" href="view/style.css">
        <!-- fevicon -->
        <link rel="icon" href="images/fevicon.png" type="image/gif" />
```

```
<!-- Scrollbar Custom CSS -->
<link rel="stylesheet" href="css/jquery.mCustomScrollbar.min.css">
<!-- Tweaks for older IEs-->
<link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css">
<!-- fonts -->
<link href="https://fonts.googleapis.com/css?family=Poppins:400,700|Raleway:400,700&display=swap" rel="stylesheet">
<!-- owl stylesheets -->
<link rel="stylesheet" href="css/owl.carousel.min.css">
<link rel="stylesheet" href="css/owl.theme.default.min.css">
<link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<link rel='stylesheet' href='https://rawgit.com/FortAwesome/Font-Awesome/master/css/font-awesome.min.css'>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css" media="screen">
</head>
<%
    if (request.getParameter("success") != null) {%>
<script>alert('Detection and Analysis Completed');</script>
<%}
%>
```

```
<body>
   <!--header section start -->
   <div class="header_section">
      <div class="container-fluid">
         <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <div class="logo"><a href=""></a></div>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
               <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
               <ul class="navbar-nav mr-auto">
                  <li class="nav-item" >
                     <a class="nav-link" href="Home.jsp">Home</a>
                  </li>
                  <li class="nav-item" style="width: 200px ">
                     <a class="nav-link" href="uploaddata.jsp">Upload Dataset</a>
                  </li>
                  <li class="nav-item" style="width: 200px ">
                     <a class="nav-link" href="LoadedData.jsp">Data Cleaning</a>
                  </li>
```

```
                <li class="nav-item" style="width: 200px ">
                  <a        class="nav-link"       href="DataCleaning.jsp">Pre-
Processing</a>
                </li>
                <li class="nav-item" style="width: 200px ">
                  <a        class="nav-link"       href="Detection.jsp">Racism
Detection</a>
                </li>
                <li class="nav-item" style="width: 100px ">
                  <a class="nav-link" href="Graph.jsp">Graph</a>
                </li>
                 <li class="nav-item" style="width: 100px ">
                  <a class="nav-link" href="index.jsp">Logout</a>
                </li>
              </ul>
            </div>
          </nav>
        </div>
      </div>
      <!--header section end -->
      <!-- contact section start -->
      <div class="contact_srction layout_padding">
        <div class="container">
          <h1 class="contact_taital">Racism Detection Analysis</h1>
          <br>
          <br>
```

```
<br>

<div id="DataTable">
    <div id="table_box_bootstrap"></div>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Tweet Text</th>
                <th>getSentimentScore</th>
                <th>getSentimentType</th>
                <th>getVeryPositive</th>
                <th>getPositive</th>
                <th>getNeutral</th>
                <th>getNegative</th>
                <th>getVeryNegative</th>
                <th>Annotation</th>
                <th>Label</th>
            </tr>
        </thead>
        <tbody class="scroll-pane">
            <%
                Connection con = SQLconnection.getconnection();
                Statement st = con.createStatement();
                try {
```

```
                    ResultSet rs = st.executeQuery("Select * from
senanalysisdetections");
                    while (rs.next()) {
                %>
                <tr>
                    <td ><%=rs.getString("id")%></td>
                    <td  ><%=rs.getString("tweet")%></td>
                    <td  ><%=rs.getString("getSentimentScore")%></td>
                    <td  ><%=rs.getString("getSentimentType")%></td>
                    <td  ><%=rs.getString("getVeryPositive")%>%</td>
                    <td  ><%=rs.getString("getPositive")%>%</td>
                    <td  ><%=rs.getString("getNeutral")%>%</td>
                    <td  ><%=rs.getString("getNegative")%>%</td>
                    <td  ><%=rs.getString("getVeryNegative")%>%</td>
                    <td  ><%=rs.getString("annotation")%></td>
                    <td  ><%=rs.getString("label")%></td>
                    <%                    }
                    } catch (Exception ex) {
                        ex.printStackTrace();
                    }

                    %>
                </tr>
            </tbody>
        </table>
    </div>
```

```
            </div>
   </div>
         <!-- contact section end -->
         <!-- footer section start -->
         <div class="footer_section">
            <div class="container">
            </div>
         </div>
         <!-- footer section end -->
         <!-- copyright section start -->
         <div class="copyright_section">
            <div class="container">
               <p class="copyright_text">Copyright 2023 All Rights Reserved.<a
href="">Racism Detection</a></p>
            </div>
         </div>
         <!-- copyright section end -->
         <!-- Javascript files-->
         <script src="js/jquery.min.js"></script>
         <script src="js/popper.min.js"></script>
         <script src="js/bootstrap.bundle.min.js"></script>
         <script src="js/jquery-3.0.0.min.js"></script>
         <script src="js/plugin.js"></script>
         <!-- sidebar -->
         <script src="js/jquery.mCustomScrollbar.concat.min.js"></script>
```

```
<script src="js/custom.js"></script>
<script src="view/script.js"></script>
<script src="js/owl.carousel.js"></script>
<script
src="https:cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.js">
</script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'></script>
<script src='https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.min.css'></script>
</body>
</html>
```

APPENDIX 2 – OUTPUT SCREENSHOTS

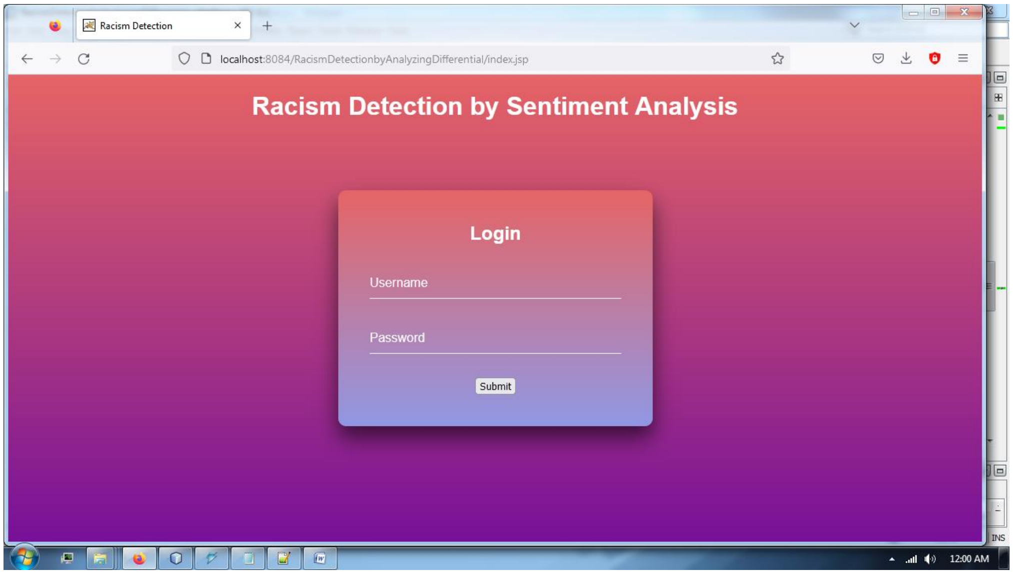


Fig 11.1: Admin Login Page

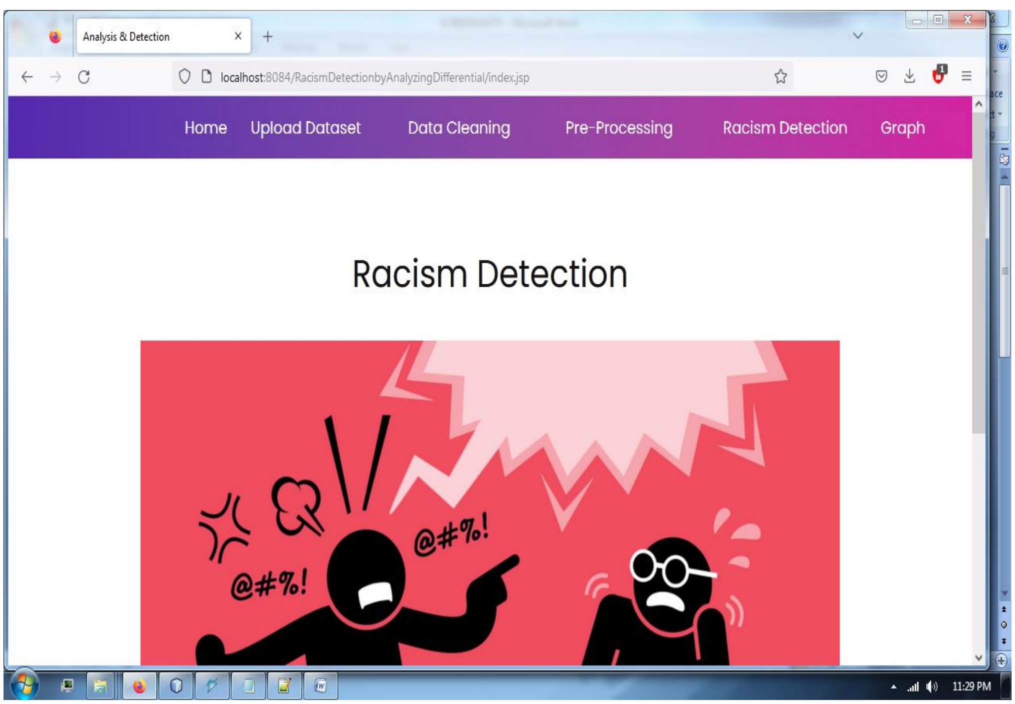


Fig 11.2: Home Page

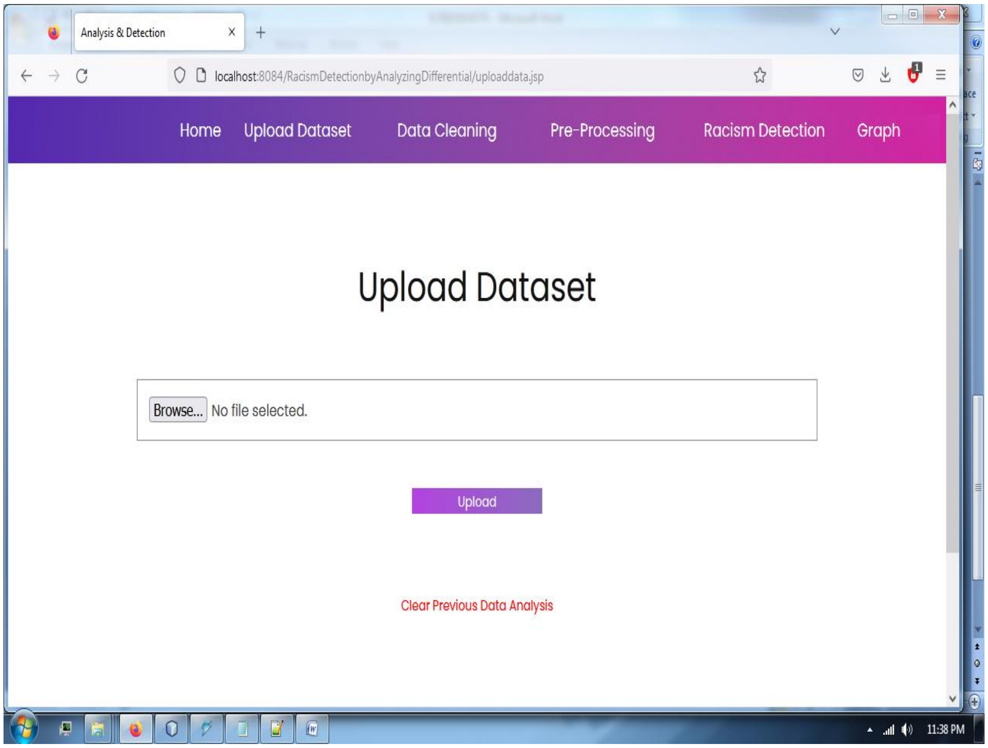


Fig 11.3: Upload Dataset

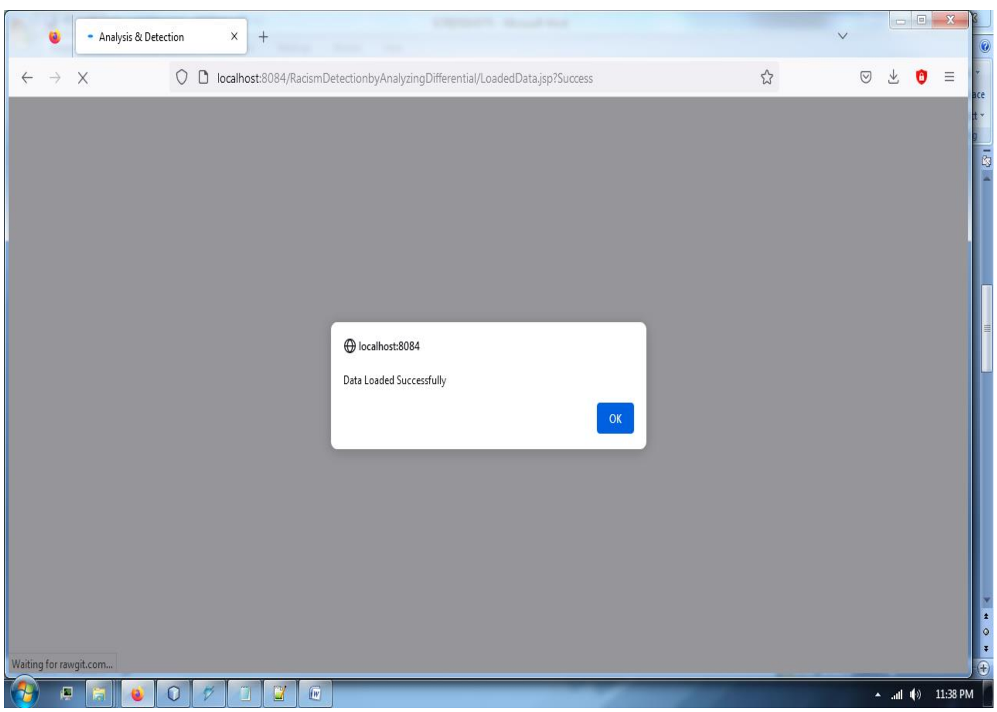


Fig 11.4: Data uploaded

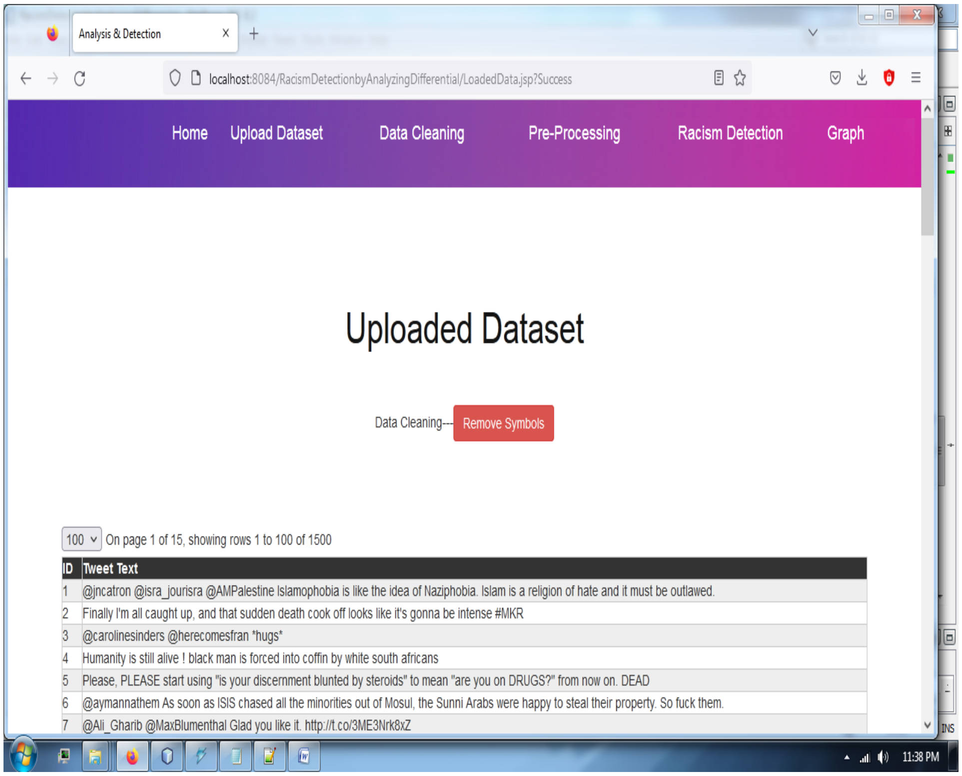


Fig 11.5: Data Cleaning

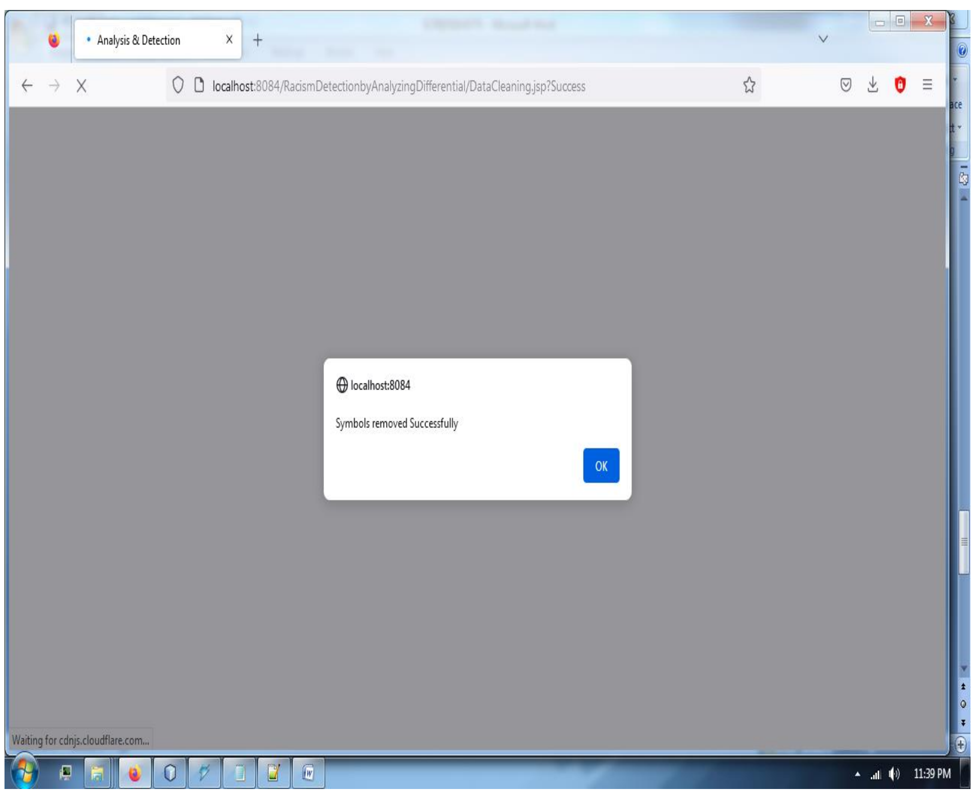


Fig 11.6: Symbols removal

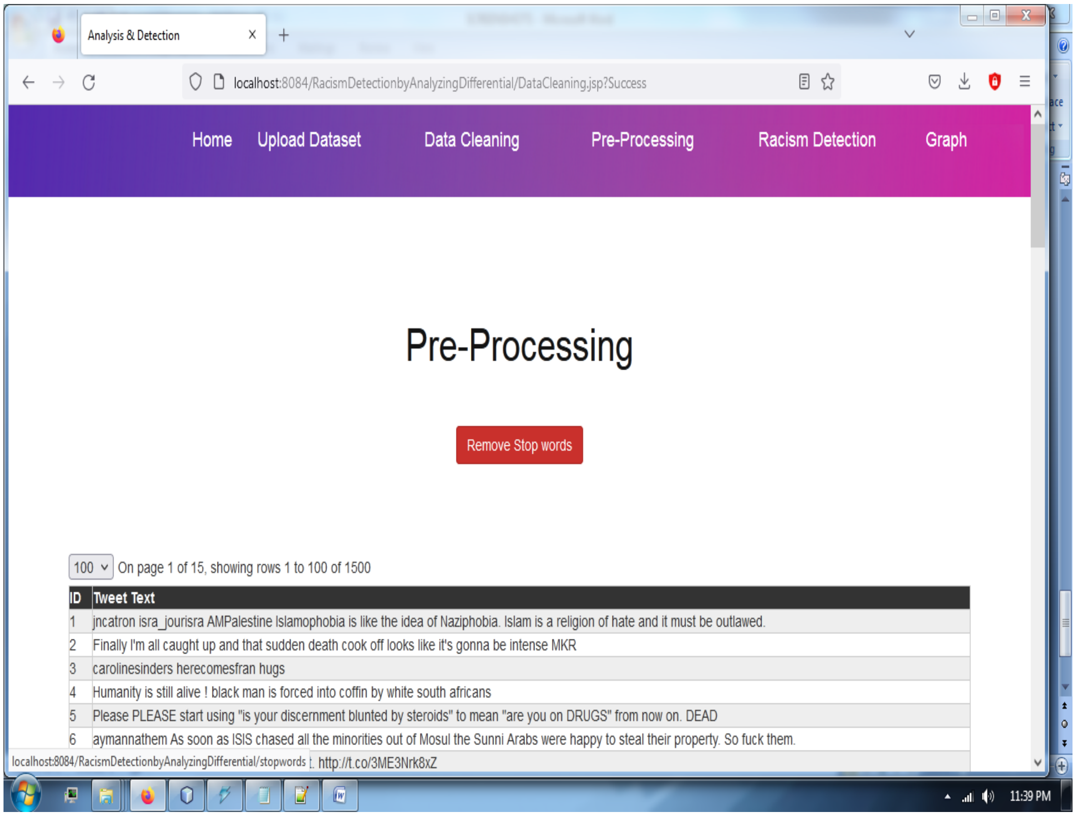


Fig 11.7: Preprocessing

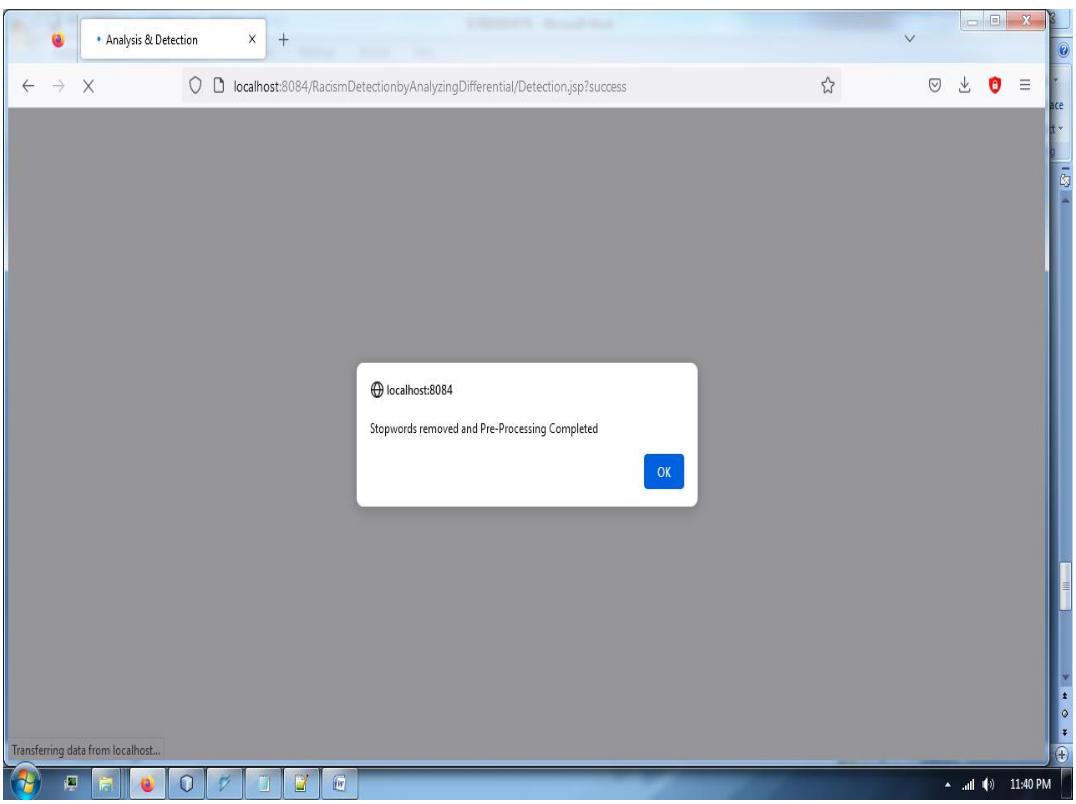


Fig 11.8: Stop words Removal

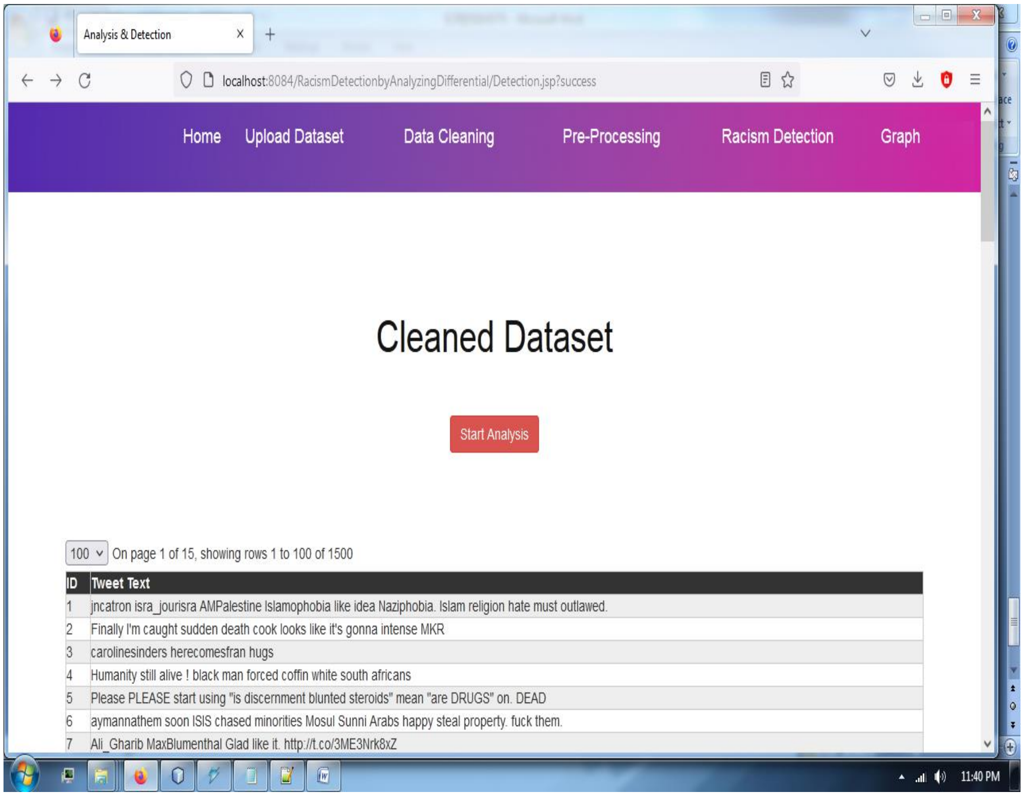


Fig 11.9: Cleaned dataset

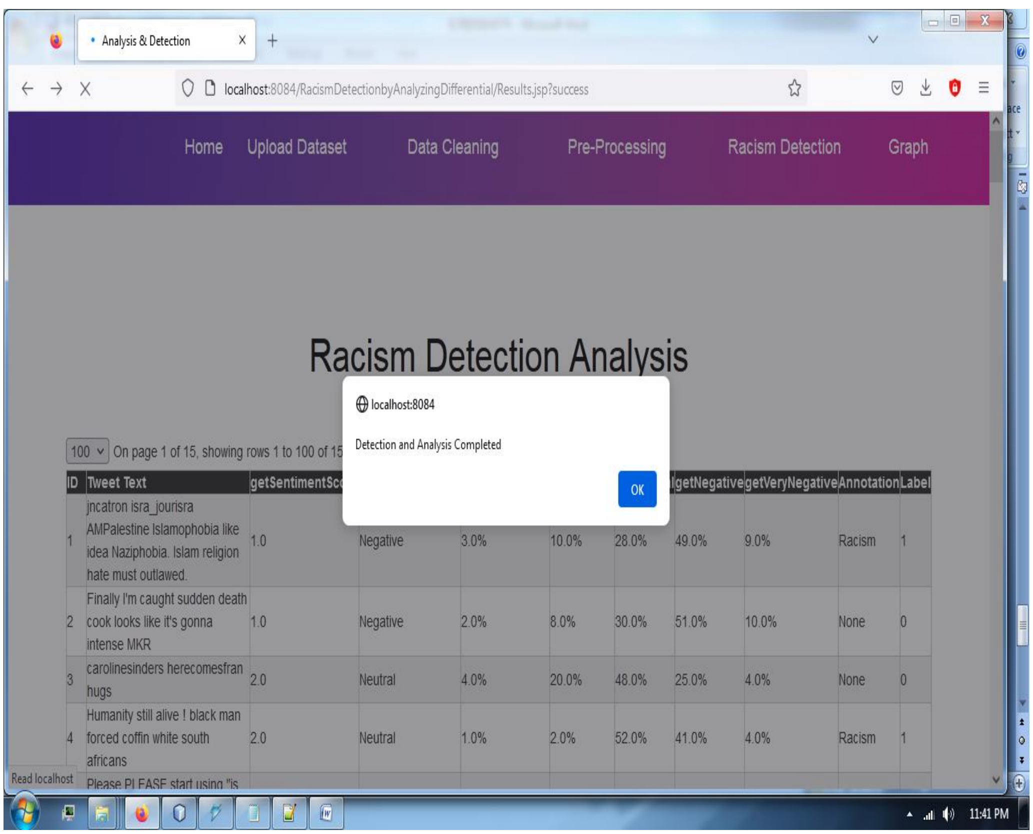
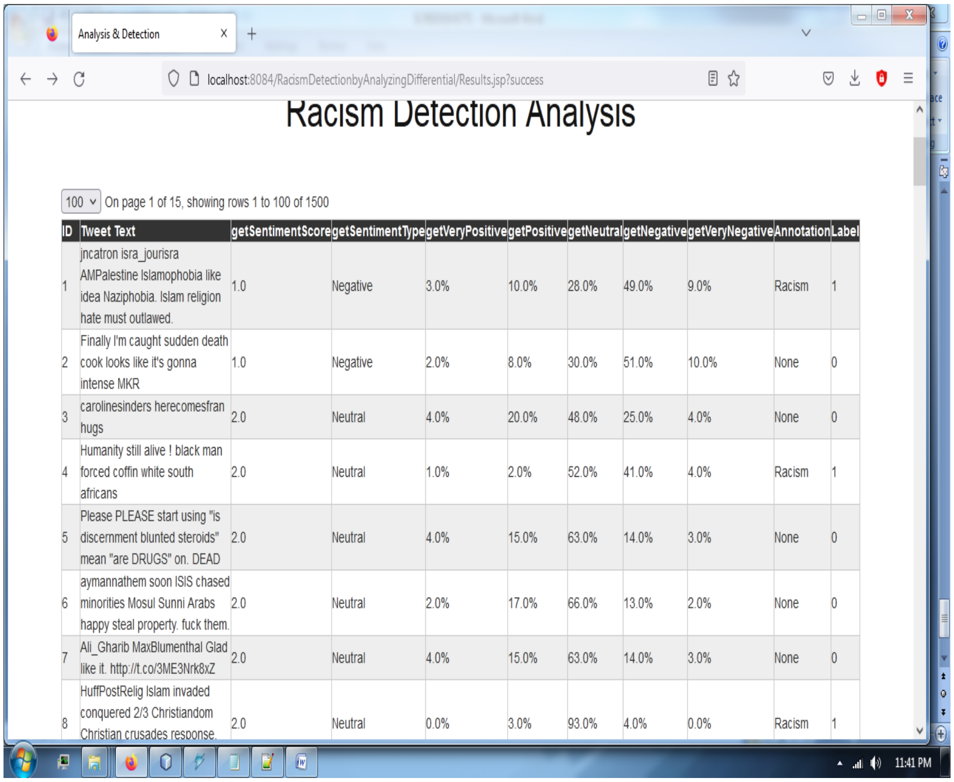


Fig 11.10: Completion of Racism detection



ID	Tweet Text	getSentimentScore	getSentimentType	getVeryPositive	getPositive	getNeutral	getNegative	getVeryNegative	Annotation	Label
1	jncatron isra_jourisra AMPalestine Islamophobia like idea Naziphobia. Islam religion hate must outlawed.	1.0	Negative	3.0%	10.0%	28.0%	49.0%	9.0%	Racism	1
2	Finally I'm caught sudden death cook looks like it's gonna intense MKR	1.0	Negative	2.0%	8.0%	30.0%	51.0%	10.0%	None	0
3	carolinesinders herecomesfran hugs	2.0	Neutral	4.0%	20.0%	48.0%	25.0%	4.0%	None	0
4	Humanity still alive ! black man forced coffin white south africans	2.0	Neutral	1.0%	2.0%	52.0%	41.0%	4.0%	Racism	1
5	Please PLEASE start using "is discernment blunted steroids" mean "are DRUGS" on. DEAD	2.0	Neutral	4.0%	15.0%	63.0%	14.0%	3.0%	None	0
6	aymannathem soon ISIS chased minorities Mosul Sunni Arabs happy steal property. fuck them.	2.0	Neutral	2.0%	17.0%	66.0%	13.0%	2.0%	None	0
7	Ali_Gharib MaxBlumenthal Glad like it. http://t.co/3ME3Nrk8xZ	2.0	Neutral	4.0%	15.0%	63.0%	14.0%	3.0%	None	0
8	HuffPostRelig Islam invaded conquered 2/3 Christiandom Christian crusades response.	2.0	Neutral	0.0%	3.0%	93.0%	4.0%	0.0%	Racism	1

Fig 11.11: Racism Detection Analysis

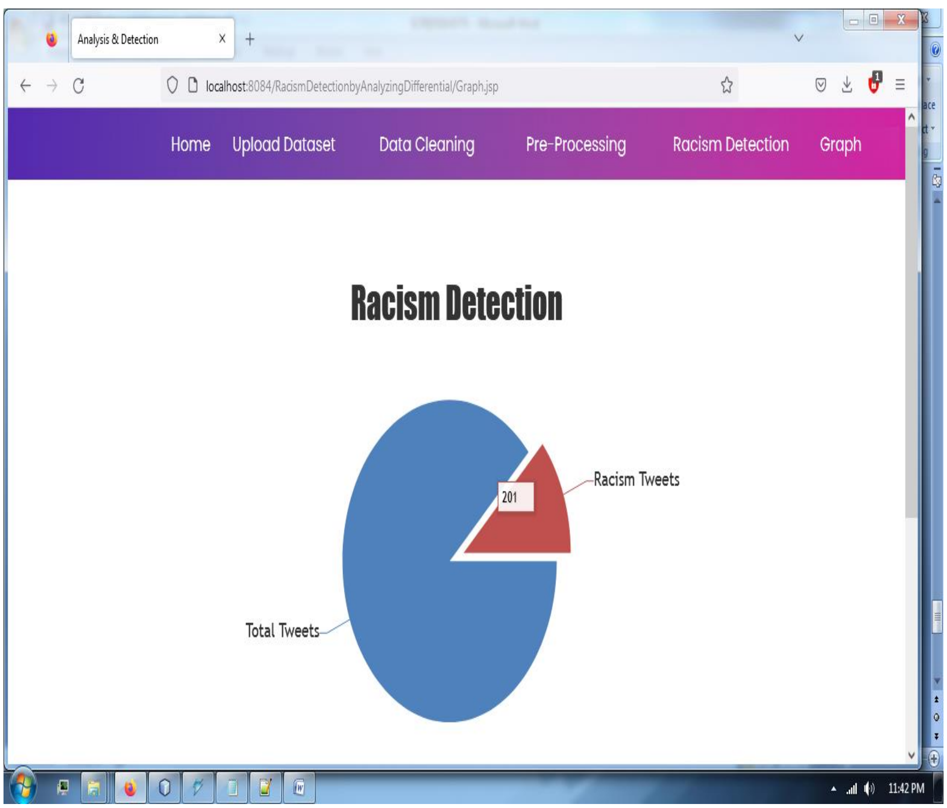


Fig 11.12: Pictorial representation

REFERENCES

1. Arango, C., Han, S., Srivastava, M., & Cambria, E. (2021). Inclusive AI: ‘A Review and Reflection on the AI Literature through the Lens of Diversity, Intersectionality, and Racism’. arXiv preprint arXiv:2102.08602.

2. Bernal, E. A., Ghaffary, M., & Huang, K. (2020). ‘Towards Detecting Offensive Memes: A Benchmark Dataset and Baselines’. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 4591-4597).

3. Bhagat, A., & Gupta, S. (2020). ‘Deep learning-based approach for cyberbullying and hate speech detection in social media texts’. Social Network Analysis and Mining, 10(1), 1-17.

4. Burnap, P., & Williams, M. L. (2015). ‘Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making’. Journal of Big Data, 2(1), 1-18.

5. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). ‘Automated Hate Speech Detection and the Problem of Offensive Language’. In Proceedings of the 11th International AAAI Conference on Web and Social Media (pp. 512-515).

6. Dinakar, K., Reichart, R., & Lieberman, H. (2012). ‘Modeling the detection of textual cyberbullying’. In Proceedings of the 5th International Conference on Weblogs and Social Media (pp. 128-135).

7. Fortuna, P., Nunes, R., & Rodrigues, J. M. (2018). ‘A survey on automatic detection of hate speech in text’. ACM Computing Surveys (CSUR), 51(4), 1-30.

8. García-Gómez, J., Gil, D., Molina-González, M. D., García-Serrano, A., & Araque, O. (2020). ‘Automatic detection of hate speech in Twitter conversations using ensemble learning’. Information Processing & Management, 57(2), 102082.

9. Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). ‘Word embeddings quantify 100 years of gender and ethnic stereotypes’. Proceedings of the National Academy of Sciences, 115(16), E3635-E3644.

10. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., ... & Smith, N. A. (2011). ‘Part-of-speech tagging for Twitter: Annotation, features, and experiments’. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Vol. 1, pp. 42-47).

11. Kwok, I., & Wang, Y. (2013). ‘Locate the hate: Detecting tweets against blacks’. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1300-1309).

12. Lai, C. H., Liu, K. Y., Lee, S. J., Lee, W. H., & Wu, C. H. (2020). ‘Hate speech detection on social media using sentiment analysis and CNN’. Expert Systems with Applications, 155, 113500.

13. Malmasi, S., & Zampieri, M. (2018). ‘Benchmarking language identification and POS-tagging for low-resource languages’. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

14. Olteanu, A., Castillo, C., Boy, J., & Varshney, K. R. (2018). ‘The effect of extremist violence on hateful speech online’. In Proceedings of the 2018 World Wide Web Conference (pp. 619-628).

15. Park, S., & Fung, P. (2017).’ One-step and two-step classification for abusive language detection on Twitter’. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 2, pp. 195-202).

16. Pellegrini, T., Aranha, M., Marinho, L. B., & Melo, P. O. (2017). ‘Automatic racism detection on Twitter: A text mining approach. Information Processing & Management’, 53(1), 255-268.

17. Ribeiro, M. H., Calais, P. H., Santos, Y., & Silva, F. M. (2020).’ Automatic hate speech detection: A survey. Journal of Inforsmation and Data Management’, 11(2), 371-399.

18. Ross, B., Katakwar, P., & Kumar, R. (2020). ‘A Survey on Hate Speech Detection using Natural Language Processing’. arXiv preprint arXiv:2005.12943.

19. Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). ‘Recursive deep models for semantic compositionality

over a sentiment treebank'. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631-1642).

20. Waseem, Z., & Hovy, D. (2016). 'Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter'. In Proceedings of the NAACL Student Research Workshop (pp. 88-93).