**ENHANCING SOCIAL MEDIA SENTIMENT ANALYSIS WITH EMOJI AND EMOTICON EMBEDDINGS FOR MITIGATING ONLINE TOXICITY**

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

## Bachelor of Technology in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning

*by*

**PRANSHU CHOUBEY (20BAI1069)**



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

April, 2024



**DECLARATION**

I hereby declare that the thesis entitled “Enhancing Social Media Sentiment Analysis with Emoji and Emoticon Embeddings for Mitigating Online Toxicity” submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning, Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Dr Sandhya M.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: **Signature of the Candidate**



**School of Computer Science and Engineering**

CERTIFICATE

This is to certify that the report entitled **“Enhancing Social Media Sentiment Analysis with Emoji and Emoticon Embeddings for Mitigating Online Toxicity”** is prepared and submitted by **Pranshu Choubey (20ABI1069)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. Sandhya M

Date:

Signature of the Internal Examiner Signature of the External Examiner

Name: Name:

Date: Date:

Approved by the Head of Department,

**B.Tech. CSE with Specialization in Artificial Intelligence and Machine Learning**

Name: Dr. Sweetlin Hemalatha C

Date

(Seal of SCOPE)

**ABSTRACT**

Social media platforms have become vital channels for individuals to express themselves and engage in communication on a global scale. However, along with the benefits of connectivity, they also harbor significant challenges, particularly concerning the proliferation of online toxicity and hate speech. The prevalence of such harmful content underscores the urgent need for effective moderation strategies to cultivate a safer and more inclusive digital environment.

Traditional sentiment analysis methods, while useful in identifying positive and negative sentiments in text data, often struggle to accurately capture the nuances of online communication. One significant limitation lies in their inability to adequately account for the contextual information conveyed by emojis and emoticons. These visual elements play a crucial role in augmenting textual content with emotional cues, adding layers of meaning that are essential for understanding the true sentiment behind a message.

The proposed Capstone Project aims to address this gap by developing an innovative methodology that integrates emoji and emoticon embeddings into hate speech detection models. By leveraging these visual cues alongside textual data, the project seeks to enhance the accuracy and effectiveness of sentiment analysis in identifying and flagging toxic content on social media platforms.

Central to the project is the development of a function designed to process tweets containing emojis, extracting and generating embeddings that preserve the contextual information conveyed by these visual elements. These embeddings serve as a bridge between the textual and visual components of social media messages, enabling hate speech detection models to analyze both aspects simultaneously.

The methodology involves collecting diverse Twitter data, designing a model architecture capable of integrating emoji embeddings, and conducting thorough analysis to evaluate performance. The findings hold significant implications for mitigating online toxicity and fostering healthier online discourse. Future research directions are also discussed, promising further advancements in social media moderation and sentiment analysis.

**ACKNOWLEDGEMEN****T**

It is my pleasure to express with deep sense of gratitude to Dr. Sandhya M, Professor Grade 1, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Natural Language Processing (NLP).

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor i/c & Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr. Sweetlin Hemalatha C, Head of the Department, B.Tech. CSE with Specialization in Artificial Intelligence and Machine Learning and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: **Pranshu Choube****y**

# CONTENTS

**TITLE ii**

[**DECLARATION**](#_TOC_250026) **iii**

[**CERTIFICATE**](#_TOC_250025) **iv**

[**ABSTRACT**](#_TOC_250024) **v**

[**ACKNOWLEDGEMENT**](#_TOC_250023) **vi**

**CONTENT vii**

[**LIST OF FIGURES**](#_TOC_250022) **ix**

[**LIST OF TABLES**](#_TOC_250021) **x**

[**LIST OF ACRONYMS**](#_TOC_250020) **xi**

**CHAPTER 1 - INTRODUCTION**

* 1. [INTRODUCTION 1](#_TOC_250019)
  2. [CHALLENGES](#_TOC_250018) 3
  3. [PROJECT STATEMENT](#_TOC_250017) 4
  4. OBJECTIVE 5
  5. [SCOPE OF THE PROJECT](#_TOC_250016) 6

**CHAPTER 2 - BACKGROUND**

* 1. [NATURAL LANGUAGE PROCESSING (NLP)](#_TOC_250015) 8
  2. [SENTIMENT ANALYSIS](#_TOC_250014) 9
  3. [HATE SPEECH DETECTION](#_TOC_250013) 10
  4. [EMOJIS AND EMOTICONS IN SENTIMENT ANALYSIS](#_TOC_250013) 10
  5. [LITERATURE REVIEW](#_TOC_250013) 11
  6. [CONCLUSION](#_TOC_250013) 19

**CHAPTER 3 - METHODOLOGY**

* 1. [INTRODUCTION 2](#_TOC_250012)1
  2. [DATA COLLECTION AND PRE-PROCESSING 2](#_TOC_250011)2
  3. [EXPLORATORY DATA ANALYSIS (EDA) 2](#_TOC_250011)3
  4. [FEATURE ENGINEERING 25](#_TOC_250011)
  5. [MODEL BUILDING 2](#_TOC_250011)6
  6. [MODEL EVALUATION AND OPTIMIZATION 2](#_TOC_250011)8
  7. [TESTING 2](#_TOC_250011)9

**CHAPTER 4 - PROPOSED SYSTEM**

* 1. [DATASET 3](#_TOC_250009)1
  2. [DATA VISUALIZATION 3](#_TOC_250008)3
  3. EMOJI EMBEDDING GENERATION [3](#_TOC_250007)5
  4. HASHTAG EXTRACTION [3](#_TOC_250007)7

**CHAPTER 5 – RESULTS AND DISCUSSION**

* 1. [RESULTS AND FINDINGS 42](#_TOC_250005)
  2. [PERFORMANCE METRICS 47](#_TOC_250004)
  3. [PERFORMANCE ON UNSEEN DATA 47](#_TOC_250004)

**CHAPTER 6 - CONCLUSION AND FUTURE WORK**

* 1. [CONCLUSION 55](#_TOC_250003)
  2. [FUTURE WORK 55](#_TOC_250002)

[**REFERENCES 56**](#_TOC_250001)

[**APPENDICES 56**](#_TOC_250001)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure No. | Name | Page No. |
| 1 | Applications of NLP | 8 |
| 2 | Sentiment Analysis Process | 9 |
| 3 | Architecture of a Hate Speech Detection Model | 10 |
| 4 | Emoji and Emoticon Ideograms related to various Emotion Classes | 11 |
| 5 | Exploratory Data Analysis (EDA) Process | 25 |
| 6 | TF-IDF Vectorizer in Feature Engineering | 26 |
| 7 | Training Dataset (train.csv) | 33 |
| 8 | Testing Dataset (test.csv) | 34 |
| 9 | Phases of Data Processing | 34 |
| 10 | Visualizing the Distribution of Hate and Non-Hate Tweets via Count Plot | 35 |
| 11 | Visualizing the Sentiment Distribution via a Pie Chart | 35 |
| 12 | Word Cloud: Most Frequent Words used in Non-Hate Speech Tweets | 36 |
| 13 | Word Cloud: Most Frequent Words used in Hate Speech Tweets | 36 |
| 14 | Proposed Architecture | 37 |
| 15 | Embedding Generation Function | 38 |
| 16 | Embedding Generation Sample Output | 39 |
| 17 | Hashtag Extraction Function | 41 |
| 18 | Hashtag Extraction Sample Output | 41 |
| 19 | Performance on Unseen Data – 1: Generate New Test Data | 43 |
| 20 | Performance on Unseen Data – 2: Pre-process New Test Data | 44 |
| 21 | Performance on Unseen Data – 3: Test New Test Data | 45 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table No. | Name | Page No. |
| 1 | Performance Metrics using Logistic Regression | 42 |
| 2 | Performance Metrics using Grid Search after Hyper-Parameter Tuning | 42 |

**LIST OF ACRONYMS**

| **Acronym** | **Description** |
| --- | --- |
| ALBERT | A Lite BERT |
| API | Application Programming Interface |
| AUC | Area Under the ROC Curve |
| BiLSTM | Bidirectional Long Short-Term Memory |
| BERT | Bidirectional Encoder Representations from Transformers |
| CE | Contextual Emoji-enhanced |
| CNN | Convolutional Neural Network |
| CSV | Comma-Separated Values |
| EDA | Exploratory Data Analysis |
| ESR | Emotion and Sentiment Recognition |
| FAET | Fine-grained Attention Network |
| FastText | FastText Word Embeddings |
| F1 | F1 Score |
| GloVe | Global Vectors for Word Representation |
| HTML | Hypertext Markup Language |
| JSON | JavaScript Object Notation |
| LR | Logistic Regression |
| LSTM | Long Short-Term Memory |
| NLP | Natural Language Processing |
| S-BERT | Sentence-BERT |
| SE | Software Engineering |
| SHAP | SHapley Additive exPlanations |
| SVM | Support Vector Machine |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| k-NN | k-Nearest Neighbors |

# CHAPTER 1

**Introduction**

**1.1 INTRODUCTION**

In the contemporary digital landscape, characterized by prolific social media engagement, platforms like Twitter, Instagram and Facebook serve as virtual arenas where individuals express their opinions, share information, and engage in discourse on a global scale. However, amidst the vast sea of online communication, there exists a pressing concern: the proliferation of toxic content, hate speech, and negative sentiments that permeate these platforms, particularly evident in the brevity and spontaneity of tweets. The ability to discern and mitigate such harmful expressions has become a paramount challenge for social media platforms, users, and researchers alike.

This project, titled "Enhancing Social Media Sentiment Analysis through Emoji Embeddings," delves into the intricate realm of sentiment analysis on Twitter, with a specific focus on integrating emojis and emoticons to improve the detection of hate speech and toxic content within tweets. Emojis and emoticons, once considered mere embellishments in digital conversations, have evolved into powerful tools for conveying emotions, attitudes, and sentiments within the constrained format of tweets. However, traditional sentiment analysis methods often overlook the rich contextual information encoded within these visual elements, leading to limited accuracy in detecting toxicity and negative sentiment.

The primary objective of this project is to bridge this gap by proposing an innovative methodology that harnesses the potential of emoji and emoticon embeddings to enhance sentiment analysis models tailored for Twitter data. By integrating these visual cues with textual data extracted from tweets, the project aims to develop a more nuanced understanding of online discourse, particularly in identifying and flagging hate speech and toxic content within the short and dynamic nature of tweets.

The significance of this endeavor lies in its potential to contribute to the creation of safer and more inclusive online spaces within the Twitter ecosystem. As Twitter continues to play a pivotal role in shaping public discourse and influencing societal attitudes, addressing the issue of online toxicity is imperative for fostering healthy digital environments. By leveraging advanced sentiment analysis techniques augmented with emoji embeddings, this project seeks to empower platform moderators, policymakers, and users with the tools necessary to combat hate speech and promote constructive dialogue within the unique constraints of Twitter's platform.

The journey towards achieving this goal encompasses several key components. First and foremost is the collection of diverse Twitter data, comprising tweets that span a wide range of topics, sentiments, and user demographics. This dataset serves as the foundation upon which the subsequent phases of the project are built. Next comes the design and implementation of a novel model architecture capable of integrating emoji embeddings into existing sentiment analysis frameworks tailored specifically for tweets. This phase involves leveraging machine learning and natural language processing techniques to develop a robust and scalable solution optimized for Twitter data.

Throughout the project lifecycle, emphasis is placed on rigorous analysis and evaluation to assess the effectiveness and performance of the proposed methodology within the context of Twitter. By systematically measuring key metrics such as accuracy, precision, recall, and F1 score, the project aims to provide empirical evidence of its efficacy in detecting hate speech and toxic content within tweets. Additionally, qualitative analysis of the results offers insights into the contextual nuances of online communication within the Twitter platform, shedding light on the complex interplay between language, emotion, and social dynamics.

The integration of emoji embeddings represents a promising avenue for enhancing social media sentiment analysis on Twitter and combating online toxicity within its unique format and constraints. By combining computational methods with insights from linguistics, psychology, and sociology, this project endeavors to contribute to the ongoing efforts to create a more inclusive and safer digital spaces within the Twitter ecosystem.

### 1.2 CHALLENGES

The challenges faced in the pursuit of enhancing social media sentiment analysis with a focus on identifying and flagging hate or dark comments/tweets span various aspects of natural language processing, data analysis, and the unique characteristics of social media communication. These challenges include:

1. **Dynamic Nature of Online Communication:** Social media platforms host a diverse range of users with varying linguistic preferences and cultural backgrounds, leading to a complex landscape of expressions and interpretations. This diversity poses challenges in understanding the contextual nuances of language, making it difficult to develop universal sentiment analysis models.
2. **Ambiguity of Human Language:** Sentiments expressed in text can be subtle, sarcastic, or context-dependent, making it challenging for traditional sentiment analysis approaches to capture the true intent behind messages. Hate speech and dark sentiments often manifest in veiled language, sarcasm, or implicit references, further complicating detection efforts.
3. **Complexity of Emojis and Emoticons:** While emojis and emoticons enrich communication by conveying emotions and nuances, they also present challenges for sentiment analysis. Interpretations of emojis can vary across cultures and contexts, requiring robust techniques to accurately capture their intended meanings and correlate them with textual content.
4. **Interdisciplinary Approaches:** Addressing these challenges requires interdisciplinary approaches that combine insights from natural language processing, machine learning, and computational linguistics. By leveraging advances in these fields, researchers can develop more robust sentiment analysis techniques capable of navigating the complexities of online discourse and effectively identifying hate or dark comments/tweets.
5. **Real-time Toxicity Detection and Scalability:** Designing algorithms and systems that ensure real-time processing and scalability for the efficient detection of online toxicity. Develop solutions that can handle the large volume of social media data while providing timely analysis and content moderation.
6. **Ethical Content Moderation:** Tackling ethical considerations in content moderation specific to hate speech and toxic content. Strive for a balance between responsible content regulation and respecting users' freedom of expression, emphasizing fairness, transparency, and bias mitigation.
7. **Emotion-Emoji Associations in Toxic Expressions:** Investigating the specific associations between emotions and emojis/emoticons in the context of toxic expressions. Develop models that recognize the nuanced emotional nuances conveyed by visual elements in online toxicity.
8. **Limited Labeled Datasets for Hate Speech:** Acquiring labeled datasets specifically tailored for hate speech and dark sentiments is challenging due to the sensitivity of the content. Developing effective training models necessitates overcoming the limitations of available datasets, possibly through techniques such as transfer learning or domain adaptation.

**1.3 PROBLEM STATEMENT**

**Problem:** The exponential growth of social media platforms as primary channels for communication has led to a concerning proliferation of hate speech, dark sentiments, and toxic comments within online discourse. Existing sentiment analysis models, primarily designed to interpret textual content, struggle to accurately identify and flag such harmful content due to its nuanced nature and the evolving linguistic landscape. Traditional methods overlook the rich contextual information conveyed by emojis and emoticons, leading to limited accuracy in detecting toxicity.

**Solution:** This research addresses the inadequacy of traditional sentiment analysis methods in effectively discerning hate or dark comments/tweets within the dynamic and diverse environment of social media. By integrating advanced techniques such as emojis and emoticons embeddings, this project aims to bridge this gap. The proposed methodology seeks to develop a comprehensive framework capable of accurately identifying and flagging hate or dark comments/tweets by exploring the visual and contextual dimensions of online communication.

**Impact:** The proposed solution holds the potential to significantly enhance social media sentiment analysis by leveraging visual and contextual cues. By accurately identifying and flagging hate or dark comments/tweets, the developed framework can contribute to fostering healthier online environments and mitigating the spread of toxic discourse. This research not only addresses a pressing issue in the digital age but also offers a pathway towards more effective content moderation strategies on social media platforms.

**1.4 OBJECTIVES**

The primary objective of this research project is to advance the field of sentiment analysis within the context of social media by leveraging emojis and emoticons embeddings to specifically address and mitigate online toxicity. The following research objectives outline the key goals and milestones for achieving success in this endeavor:

1. **Develop Innovative Methodology:** Design and implement a novel methodology for enhancing sentiment analysis on social media platforms by integrating emoji and emoticon embeddings into hate speech detection models. This includes creating embeddings specifically tailored for emojis and emoticons to enhance the emotional awareness of sentiment analysis models and address the emotive ambiguity inherent in emojis and emoticons.
2. **Generate Embeddings:** Create a function capable of processing tweets containing emojis, generating embeddings to convert text+emoji tweets into text-only while preserving emoji context. This involves developing techniques to disambiguate visual elements when conveying hate, dark sentiments, or other forms of online toxicity.
3. **Improve Sentiment Analysis Models:** Utilize the generated embeddings to enhance existing sentiment analysis models, thereby improving their accuracy in detecting hate speech and toxic content. Enhancements include refining algorithms for real-time processing, scalability, and integrating contextual sensitivity for better interpretation of hate speech within the broader context of social media conversations.
4. **Enhance Toxicity Detection Models:** Improve existing sentiment analysis models to better identify and flag instances of online toxicity, hate speech, and dark sentiments by incorporating both textual and visual elements. This involves exploring multimodal fusion strategies to seamlessly integrate textual and visual features for a more comprehensive understanding of toxic expressions.
5. **Evaluate Effectiveness:** Conduct thorough evaluation and analysis of the developed methodology to assess its effectiveness in identifying and flagging hate or dark comments/tweets on social media. Additionally, identify potential avenues for future research and development in the field of social media sentiment analysis, including the exploration of additional modalities and advanced techniques for further enhancing detection capabilities.

**1.5 SCOPE OF THE PROJECT**

Focus on Social Media Platforms: This project will primarily focus on analyzing content from social media platforms, with a specific emphasis on Twitter. The methodology developed will be tailored to handle the unique characteristics of tweets, including their limited length and the prevalence of emojis and emoticons.

Integration of Emoji and Emoticon Embeddings: The scope of this project includes the integration of emoji and emoticon embeddings into existing sentiment analysis models. Emojis and emoticons play a crucial role in conveying sentiment and contextual information in online communication, and this project aims to leverage their potential to enhance sentiment analysis accuracy.

Hate Speech Detection and Sentiment Analysis: The project will target the identification and flagging of hate speech, dark sentiments, and toxic comments within social media content. By improving hate speech detection and sentiment analysis capabilities, the developed methodology will contribute to fostering healthier online discourse and mitigating the spread of harmful content.

Model Evaluation and Analysis: A significant aspect of this project involves evaluating the effectiveness of the developed methodology. This includes conducting rigorous testing and analysis to assess its performance in accurately identifying and classifying hate or dark comments/tweets. The evaluation process will provide insights into the strengths and limitations of the proposed approach.

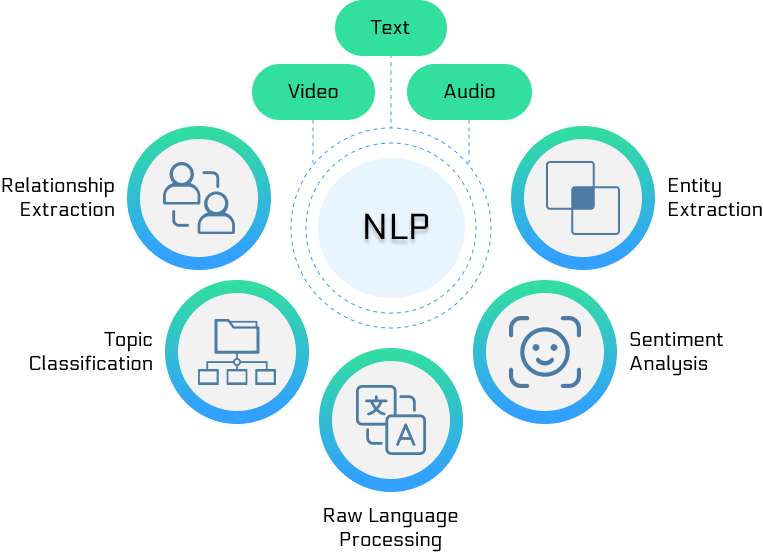
Exploration of Future Research Directions: While this project will address specific challenges related to social media sentiment analysis, its scope also encompasses the identification of potential future research directions. This may include exploring additional modalities, such as images and audio, and incorporating advanced techniques to further enhance detection capabilities and adaptability to evolving linguistic landscapes.

# CHAPTER 2

## BACKGROUND STUDIES

# 2.1 NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) is an interdisciplinary domain focused on the interaction between computers and human languages. It involves various methods and approaches to enable computers to comprehend, interpret, and produce human language in a way that is meaningful and contextually appropriate. Important aspects of NLP encompass text processing, syntactic analysis, semantic comprehension, and discourse modeling. A solid grasp of the fundamental concepts in NLP is crucial for building robust sentiment analysis models and systems to detect hate speech effectively.



# Figure 1: Applications of NLP

# 2.2 SENTIMENT ANALYSIS

# Sentiment analysis, also referred to as opinion mining, involves a natural language processing (NLP) task focused on identifying the sentiment or subjective expressions within text. Conventional sentiment analysis methods predominantly concentrate on evaluating textual material to categorize sentiments as positive, negative, or neutral. These strategies frequently utilize machine learning methods like Support Vector Machines (SVM), Naive Bayes, and neural networks for classification.

# While traditional sentiment analysis models have achieved significant success in various domains, they often struggle to capture the nuanced emotional context conveyed by emojis and emoticons. Emojis are pictorial representations used to express emotions, attitudes, and context in digital communication, while emoticons are textual representations of facial expressions. The incorporation of emojis and emoticons into sentiment analysis models has the potential to enhance their accuracy and effectiveness, particularly in social media contexts where these symbols are prevalent.

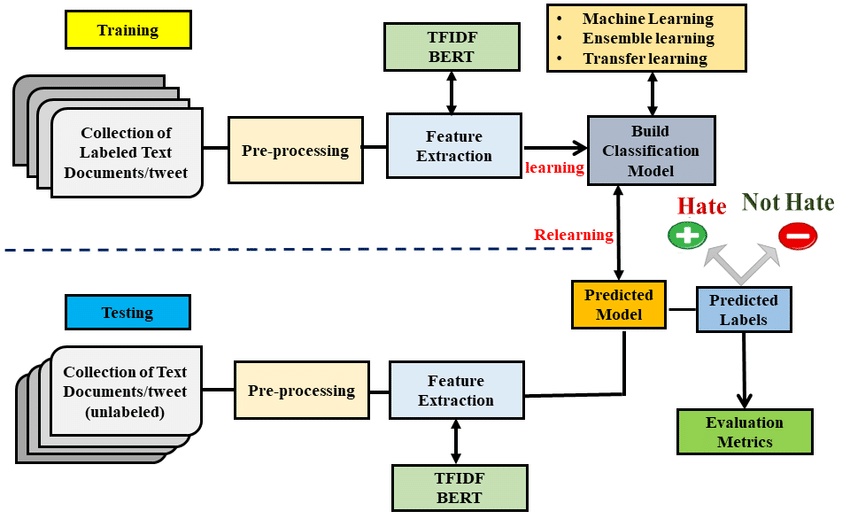
# 

# Figure 2: Sentiment Analysis Process

**2.3 HATE SPEECH DETECTION**

Hate speech detection is a critical task in online content moderation, aimed at identifying and mitigating harmful or abusive language targeted at individuals or groups based on attributes such as race, gender, religion, or sexual orientation. Hate speech detection models typically employ NLP techniques to analyze text and classify content as hate speech or non-hate speech.

Existing hate speech detection approaches often rely on lexical and syntactic features to identify discriminatory language patterns. However, these methods may overlook the nuanced contextual information conveyed by emojis and emoticons, leading to limitations in the detection accuracy, especially in social media environments where emojis are extensively used to convey sentiments and attitudes.

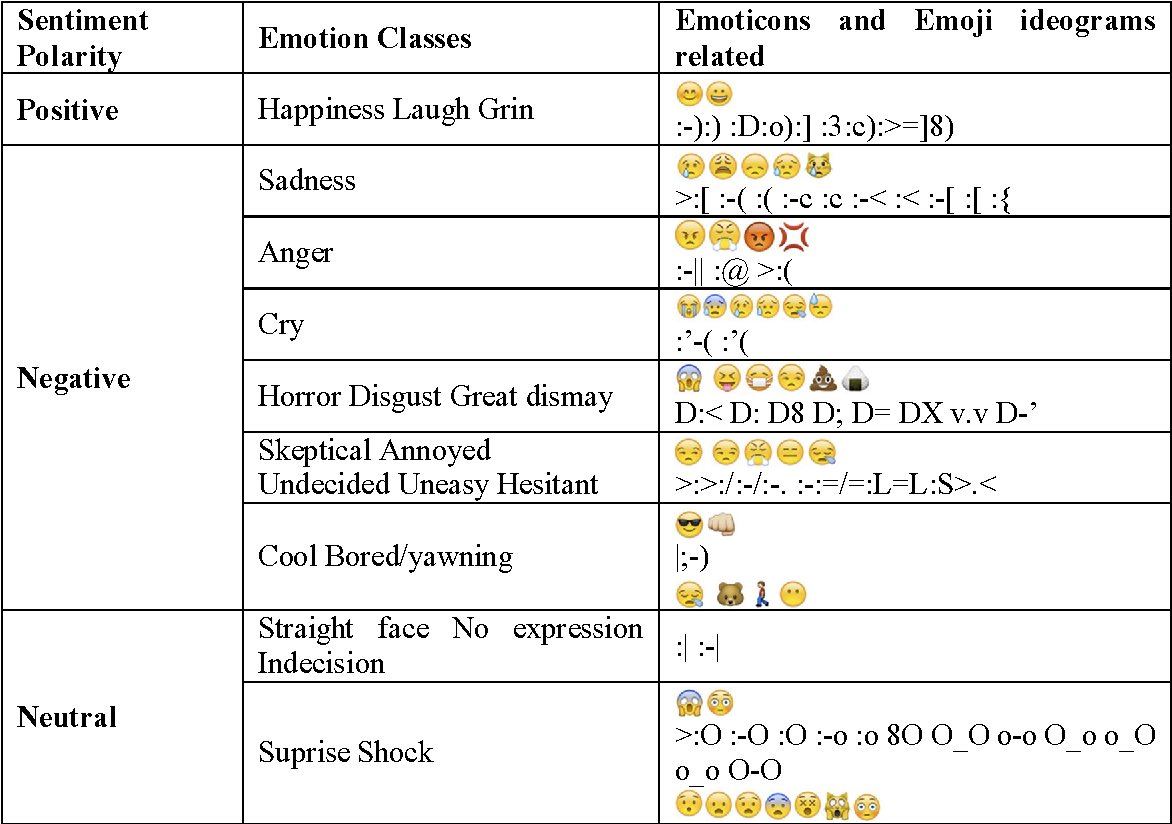


**Figure 3:** Architecture of a Hate Speech Detection Model

**2.4** **EMOJIS AND EMOTICONS IN SENTIMENT ANALYSIS**

The integration of emojis and emoticons into sentiment analysis models presents an opportunity to enrich the analysis by capturing the emotional context and nuances expressed in social media content. Recent studies have explored various approaches to incorporate emojis and emoticons into sentiment analysis, including:

* **Emoji Embeddings:** Embedding-based approaches represent emojis as dense vector representations in the same semantic space as words, allowing models to learn the contextual relationships between emojis and textual content. These embeddings can be learned from large text corpora or pretrained models and subsequently integrated into sentiment analysis architectures.
* **Multimodal Fusion:** Multimodal sentiment analysis techniques combine textual features with visual features extracted from emojis or emoticons to create a more comprehensive representation of sentiment. These approaches leverage both textual and visual cues to enhance sentiment classification performance, particularly in social media contexts where emojis play a significant role in conveying emotions.

****

**Figure 4:** Emoji and Emoticon Ideograms related to various Emotion Classes

**2.5 LITERATURE REVIEW**

Liu et al. [1] investigated the utilization of emojis in sentiment analysis of online Chinese texts, with the objective of enhancing accuracy and comprehending sentiment changes amidst the COVID-19 pandemic. The research presents CEmo-LSTM, an enhanced emoji-embedding model utilizing Bi-LSTM, and assesses its efficacy through various sentiment analysis techniques, encompassing rule-based and classification algorithms. The study contrasts the performance of CEmo-LSTM with prevalent sentiment analysis models such as LSTM and BERT across diverse experimental setups. Emphasizing the influence of emojis on sentiment analysis algorithms, the research examines different emoji usages' effectiveness in texts. Additionally, it evaluates the pandemic's impact on individual moods employing the CEmo-LSTM algorithm. The study reveals emojis' efficacy in augmenting sentiment analysis algorithm accuracy and underscores the pandemic's adverse effect on individual sentiments, culminating in a surge of passive emotions. Proposing a pioneering emoji-embedding algorithm grounded in emoji usage for sentiment analysis, the research advocates further investigation into emoji usage patterns and contexts to refine the CEmo-LSTM algorithm's performance.

Surikov et al. [2] proposes employing a word-vector and emoji/emoticon embedding model for sentiment analysis on concise informal texts sourced from social media platforms. The research utilizes a dataset comprising emotional indicators (emojis and emoticons) with associated textual meaning labels like "happiness" or "sadness". A word-vector model is constructed using a Twitter corpus to generate embeddings for both words and emojis/emoticons. These embeddings are unified within a single semantic space, and a logistic regression classifier is trained on this combined embedding model to forecast sentiment labels. The model achieves an average accuracy of 71.5% for two sentiment categories (positive and negative), surpassing baselines that lack emoji/emoticon embeddings. Performance assessment employs metrics including accuracy, F1 score, and area under the ROC curve (AUC) on a Twitter test dataset. The study concludes that the ensemble approach of integrating word and emoji/emoticon embeddings proves effective for sentiment analysis on succinct informal texts from social media.

Ayvaz et al. [3] examined the utilization of emoji characters on Twitter and their impact on sentiment analysis and text mining. Twitter serves as the data source due to its capability for real-time access to public opinions. Data is gathered for two events - New Year's Eve (positive) and the Istanbul attack (negative) - and subjected to analysis both with and without considering emojis. An R-based sentiment analysis application is developed to evaluate tweets using the SentiWordNet sentiment dictionary, which comprises positive and negative English words. Emojis are incorporated into the dictionaries based on their sentiment scores. Inclusion of emojis results in a 2% increase in positive tweet scores for the positive event. Conversely, for the negative event, emojis lead to a decrease of 1.4% in neutral tweets and a slight increase in positive tweets. The study concludes that incorporating emojis in sentiment analysis yields higher sentiment scores compared to disregarding emojis, showcasing how social media data coupled with emojis can enrich sentiment analysis endeavors.

Vashisht et al. [4] conducted a comprehensive review of trends in emoticon (emoji)-based sentiment analysis research spanning from 2005 to 2019. Various studies utilizing emoticons for sentiment classification across tweets, blogs, and other textual sources were examined. Common machine learning algorithms such as Naive Bayes, maximum entropy classifiers, SVMs, k-NN, and SGD were employed. Features encompassed emoticons, word unigrams and bigrams, part-of-speech tags, and sentiment lexicons. Metrics for performance evaluation included accuracy, precision, recall, and F1 score. Some studies reported achieving accuracy rates of up to 92-94% for sentiment classification utilizing emoticons. Identified challenges involved the absence of a standardized emoticon lexicon, ambiguity in emoticon interpretations, scarcity of emoticons in concise texts, and the inadequacy of n-grams in effectively modeling emojis. The study concluded that integrating emoticons with existing machine learning and lexicon-based approaches can enhance sentiment analysis accuracy compared to employing those techniques independently. Nonetheless, further research is warranted to tackle the identified challenges.

Guibon et al. [5] delved into the definitions of emojis and emoticons, categorizing various usages of emojis within context, such as sentiment enhancement and modification. The paper explored applying different natural language processing and machine learning techniques for sentiment analysis on texts containing emojis, including n-gram models and SVM classifiers. A proposed methodology utilized an existing sentiment lexicon (ESR) to initially detect the sentiment of emojis. Subsequently, tools like SentiStrength were employed to detect the sentiment of sentences, followed by a comparison to identify the type of emoji usage. Models were developed to automatically discern emoji usage types on annotated corpora, aiming to enhance sentiment analysis by considering different impacts of emojis based on their usage. Potential applications discussed encompassed utilizing identified emoji meanings to empower emotional conversational agents and avatars. The paper provided relevant background on existing work analyzing emoticon usage for sentiment analysis, suggesting potential extensions to emojis.

Wolny et al. [6] explored the utilization of emoticons and emoji ideograms for sentiment analysis on Twitter data, acknowledging the challenges posed by the informal language prevalent on social media platforms. Existing sentiment analysis methods primarily focus on polarity (like/dislike), yet human emotions exhibit greater complexity. Emoticons and emojis serve to capture this nuanced emotional spectrum. Twitter data was collected using Twitter APIs and pre-processed to filter tweets based on relevant hashtags and user mentions. The Python library Tweepy facilitated the connection to the Twitter Streaming API for downloading tweets in JSON format, subsequently converted to CSV or database formats. Different emoticons and emoji symbols were categorized into emotion classes such as happiness, sadness, and anger based on existing literature on basic emotions. The frequency of various emoticons/emoji in tweets was analyzed to discern sentiment and conduct multi-class emotion classification using machine learning algorithms. Incorporating emoticons and emoji analysis alongside natural language processing significantly enhanced precision in identifying various emotions expressed in tweets.

Jagadishwari et al. [7] aimed to detect sentiments expressed in social media posts employing machine learning models. The utilized dataset comprised 300,000 tweets with sentiment scores and text, split into 70% for training and 30% for testing. Data preprocessing steps encompassed removing punctuation, HTML tags, stopwords, lemmatization, and lowercasing. Four machine learning algorithms: Bernoulli Naive Bayes, Multinomial Naive Bayes, Linear Regression, and SVM were implemented. Initially, models were trained solely with text, followed by training with both text and emoticons to assess the emoticons' impact. Performance evaluation utilized classification reports, measuring accuracy, precision, recall, and F1 score. The Naive Bayes models, particularly Multinomial Naive Bayes, exhibited the best performance, achieving up to 89% accuracy. Emoticons were observed to have a negligible effect on model performance. The paper concluded that the Naive Bayes family of classifiers, particularly Multinomial Naive Bayes, are well-suited for sentiment analysis tasks based on their performance in this study.

Bhaskaran et al. [8] propose a methodology for detecting toxic online conversations utilizing a multi-task learning approach with convolutional neural networks (CNNs). The dataset comprises Wikipedia comments annotated for toxicity. Pre-processing involves text normalization, tokenization, and padding. The model architecture comprises an embedding layer, CNN layer, max-pooling layer, and softmax output layer. Training is conducted with categorical cross-entropy loss. In addition to toxicity detection, the model predicts comment length and identifies personal attacks as auxiliary tasks within a multi-task learning setup. The model reached an F1 score of 0.89 for toxicity detection, outperforming single-task baselines. Including auxiliary prediction tasks enhances the main toxicity detection task. The methodology demonstrates the effectiveness of multi-task learning with CNNs for online toxicity detection. Additionally, the incorporation of auxiliary tasks aids in capturing more context about problematic conversations.

Zou et al. [9] propose a method to enhance short text sentiment classification by incorporating emojis and emoticons. The study selected 80 emojis (40 positive, 40 negative) and utilized Word2vec to obtain vector representations of emojis and emoticons. A convolutional neural network (CNN) model was developed for classifying short texts. It accepts both text and corresponding emoji/emoticon vectors as input. Three models were compared: CNN solely on text, CNN on text with emoji embedding as additional input, and CNN on text with both emoji and emoticon embeddings. The model integrating both emoji and emoticon embeddings achieved the highest performance, with an accuracy of 84.5% on a Twitter dataset. These outperformed models utilizing only text or text with only emoji embeddings. Incorporating visual and semantic information from emojis and emoticons can effectively improve short text sentiment classification, particularly beneficial for noisy social media texts.

Sakode et al. [10] present a methodology for detecting hate speech and offensive language on Twitter employing machine learning classifiers like Logistic Regression (LR), Support Vector Machine (SVM), and Bidirectional Long Short-Term Memory (BiLSTM). The study utilizes a dataset comprising over 16,000 tweets annotated as hateful, offensive, or clean. Pre-processing involves text normalization, tokenization, and the removal of usernames and URLs. Word embeddings obtained from fastText are utilized as features for the classifiers. The BiLSTM model utilizes these embeddings as inputs. The Logistic Regression model achieves the best performance, with F1-scores of 0.78 for hate speech detection and 0.80 for offensive language detection. The BiLSTM model performs comparably well, with F1-scores of 0.76 and 0.79, respectively. It excels in capturing context from sequences of words compared to LR. The methodology demonstrates the effectiveness of various machine learning algorithms such as LR, SVM, and BiLSTM for Twitter hate speech and offensive language detection utilizing pre-trained word embeddings.

Ayvaz et al. [11] investigate the influence of emojis on sentiment analysis of tweets using machine learning classifiers such as Support Vector Machine (SVM) and Long Short-Term Memory (LSTM). The study utilizes a dataset comprising over 15,000 tweets annotated with sentiment labels and emoji presence indicators. Preprocessing steps include tokenization, lemmatization, and removal of stopwords, URLs, and user mentions. Word embeddings are obtained from GloVe and FastText models trained on tweets, while emoji embeddings are obtained separately through emoji2vec. SVM and LSTM classifiers are trained on various features, including word embeddings, emoji embeddings, their concatenation, and the presence of emojis as additional features. The study demonstrates that including emoji embeddings enhances sentiment analysis performance, with SVM+emoji2vec achieving the highest accuracy of 81.6%. Moreover, emoji presence as a separate feature also improves the models, with LSTM+emoji presence yielding 3.2% higher accuracy than the baseline. The findings suggest that emojis contain valuable sentiment cues, and their incorporation significantly benefits social media sentiment analysis.

Yang et al. [12] propose a fine-grained attention network (FAET) designed to determine the sentiment polarity of microblog texts containing both plain text and emojis. FAET learns text embeddings using ALBERT and obtains inter-emoji embeddings utilizing an attention network. It then integrates the text and emoji embeddings into the fine-grained attention network. FAET incorporates two attention mechanisms: one calculates attention weights between text and each emoji (Text2Emoji), and the other computes attention weights between each text word and emojis (Emoji2Text). The attended vectors from each attention mechanism are concatenated and fed into a CNN classifier for sentiment label prediction. Evaluation metrics include precision, recall, and accuracy. FAET outperforms baseline models that do not consider emoji semantics, such as TextCNN+word and BiLSTM+word, achieving better performance in sentiment classification accuracy. The fine-grained attention network in FAET dynamically learns features between plain text and emojis, significantly enhancing sentiment classification accuracy in microblog texts.

Velampalli et al. [13] collected Twitter data by filtering tweets using hashtags related to positive and negative sentiments, resulting in datasets containing tweets containing both text and emojis. Additionally, an existing Kaggle dataset was utilized. Sentence embedding models such as S-BERT and Universal Sentence Encoder were employed to generate fixed-length embeddings from both tweet texts and emojis, aiming to capture semantic relationships effectively. Standard neural networks and LSTM neural networks were trained on these embeddings to classify sentiment. Remarkably, the models achieved up to 98% accuracy on text sentiment classification and 70% accuracy when classifying unseen emojis. Distributed training was utilized to reduce training time by 15% compared to single-threaded training, contributing to more efficient model development. Explainable AI techniques like SHAP values were applied to examine model biases and ensure fairness in the sentiment analysis process. The researchers achieved impressive results, with up to 99.5% accuracy, 99% precision, and 99.2% recall using the Universal Sentence Encoder and LSTM model on their combined text and emoji dataset. This represented a significant improvement over prior work that relied on traditional bag-of-words models. The adoption of distributed training and newer embedding techniques played pivotal roles in enhancing the performance of sentiment analysis models, showcasing the potential of integrating text and emoji data for sentiment classification tasks.

Kavya et al. [14] explores leveraging emojis and emoticons to augment sentiment analysis on social media platforms like Twitter, known for its informal and concise nature. The proposed methodology involves gathering Twitter data using the Twitter API and converting it into a suitable format for analysis. Features like user mentions and hashtags aid in data filtering and classification. Emojis and emoticons are recognized as crucial sentiment indicators that directly convey emotions, transcending language barriers. Tweets are categorized into positive, negative, neutral, or garbage categories based on the presence and nature of emojis. A model is trained on a subset of manually classified tweets and evaluated on a separate sample to assess accuracy. The integration of natural language processing with symbol analysis is highlighted as a promising avenue for sentiment analysis enhancement. While platforms like Twitter offer vast amounts of data for opinion mining, traditional text-based methods may not suffice. The paper advocates for novel techniques that integrate both language and symbol analysis to effectively capture sentiment nuances in short social media texts.

LeCompte et al. [15] conducts a thorough evaluation of five state-of-the-art sentiment analysis tools—SentiStrength, SentiStrength-SE, SentiCR, Senti4SD, and SEntiMoji—across five datasets sourced from software engineering domains. The datasets encompass tweets and comments from platforms such as JIRA, Stack Overflow, code reviews, and a Java library documentation forum. Each sentiment analysis tool employs various machine learning algorithms, including LSTM neural networks and feature-based classifiers. Performance evaluation metrics encompass precision, recall, F1-score, and accuracy for both positive and negative sentiment classes across each dataset. SEntiMoji emerges as the top-performing tool overall, followed by SentiCR and SentiStrength-SE, while SentiStrength exhibits the lowest performance. Notably, datasets from JIRA and Stack Overflow are classified with the highest accuracy, whereas the Java library dataset poses the greatest challenge. The paper's comprehensive evaluation of sentiment analysis tools for software engineering texts offers valuable insights. These findings could be particularly beneficial for your capstone project focusing on enhancing social media sentiment analysis.

Chen et al. [16] aims to enhance emotion and sentiment detection in software engineering (SE) texts by harnessing emojis as indicators of affective states. Given the inadequacy of sentiment analysis tools trained on general domains for SE data, primarily due to technical terminology, and the limited size and lexicon coverage of existing SE-specific datasets, the researchers embarked on a novel approach. A substantial dataset comprising over 2 million tweets and GitHub posts containing emojis was collected. An emoji prediction model was trained utilizing text representations acquired through a neural network. The premise is that texts surrounding the same emoji will possess similar vector embeddings, thereby transferring sentiment knowledge from emoji usage. Alongside existing manually labeled SE datasets, this emoji-powered representation learning approach was employed to train final sentiment and emotion classifiers via transfer learning methods like Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks. The models underwent evaluation on benchmark SE sentiment and emotion datasets employing metrics such as macro-F1. Compared to prior techniques, the proposed approach achieved a notable average improvement of 0.036 and 0.049 in macro-F1 for sentiment and emotion detection, respectively, underscoring significant gains derived from leveraging large-scale emoji data. Notably, tweets contributed substantially, emphasizing the value of open-domain knowledge for domain-specific tasks. The paper presents an effective strategy to tackle data scarcity in SE sentiment analysis by exploiting ubiquitous signals like emojis. While performance still falls below human levels, this approach opens up new avenues for leveraging readily available web data to advance affective computing in software engineering.

**2.6 CONCLUSION**

In this literature review, we have explored a diverse range of research papers focused on enhancing sentiment analysis and addressing online toxicity through the incorporation of emojis and emoticons. These studies underscore the significance of emojis and emoticons as powerful indicators of sentiment and emotion in social media texts, particularly in platforms like Twitter where communication is often concise and informal.

From examining various methodologies and techniques proposed in the reviewed papers, it is evident that emojis and emoticons play a crucial role in augmenting sentiment analysis tasks. Embracing these visual elements enables sentiment analysis models to capture nuanced emotional cues that may not be adequately conveyed through text alone. Moreover, leveraging emojis and emoticons can significantly improve the performance of sentiment analysis models, especially in contexts like software engineering and social media, where traditional text-based methods may fall short due to domain-specific jargon or data scarcity.

Drawing insights from the reviewed literature, this project aims to address the limitations of existing sentiment analysis models by incorporating emojis and emoticons. By developing a novel function that processes tweets with emojis and generates embeddings to convert text+emoji tweets into text-only while preserving emoji context, the project endeavors to enhance hate speech detection and sentiment analysis on social media platforms like Twitter.

By integrating emojis and emoticons into sentiment analysis models, the project seeks to not only improve the accuracy of hate speech detection but also enhance existing sentiment analysis models' ability to discern subtle emotional nuances in social media texts. This innovative approach holds promise for mitigating online toxicity and fostering a safer and more inclusive online environment.

Overall, the findings from this literature review provide valuable insights and serve as a solid foundation for this project, offering a roadmap for leveraging emojis and emoticons to enhance sentiment analysis and combat online toxicity effectively. Through the integration of visual cues into sentiment analysis models, the project endeavors to contribute to the advancement of affective computing and promote positive online interactions.

# CHAPTER 3

## METHODOLOGY

**3.1 INTRODUCTION**

Top of Form

This section outlines the systematic approach employed to develop and evaluate a hate speech detection system using machine learning techniques. Detecting hate speech in online content is a complex task that requires a multi-faceted approach encompassing data preprocessing, exploratory data analysis, feature engineering, model building, evaluation, and optimization. It provides a detailed overview of each step undertaken in the development of the hate speech detection system.

The methodology begins with data collection, where a diverse dataset comprising text data from various online platforms is curated. Following this, extensive data preprocessing techniques are applied to clean and prepare the dataset for analysis. Subsequently, exploratory data analysis (EDA) is conducted to gain insights into the dataset's characteristics and distribution, providing valuable information for subsequent processing steps.

Feature engineering, a crucial step in natural language processing (NLP) tasks, follows the EDA phase. Here, the raw text data is transformed into numerical features using techniques such as TF-IDF vectorization. This process enables the conversion of textual information into a format suitable for machine learning algorithms, facilitating the model building phase.

Model building involves the selection and training of machine learning algorithms for hate speech detection. In this study, a logistic regression classifier is chosen for its simplicity and effectiveness in text classification tasks. The dataset is split into training and testing sets, and the logistic regression model is trained on the training set.

After the model is constructed, it is subjected to a thorough assessment using relevant metrics, including F1-score, accuracy, precision, and recall. A confusion matrix also serves to illustrate the model's performance and highlight areas that need work. Hyperparameter tuning techniques, including grid search with cross-validation, are employed to optimize the model's performance further.

Following model evaluation and optimization, the hate speech detection system is tested on a separate dataset to assess its generalization performance. The testing dataset comprises similar data to that used for training but has not been seen by the model during training. The pre-trained model is applied to this dataset, and its performance is evaluated using the same metrics as in the training phase.

**3.2 DATA COLLECTION AND PRE-PROCESSING**

The first step in the methodology is the collection of data, which involves obtaining a dataset that contains tweets along with their corresponding labels indicating whether they are hate speech or not. The dataset used in this project is acquired from an external source and consists of three columns: **id**, **label**, and **tweet**.

After acquiring the dataset, the next step is data preprocessing. This involves several steps to clean and prepare the text data for analysis and model training. The preprocessing steps include:

* **Lowercasing**: Convert all text to lowercase to ensure consistency in text analysis.
* **Removal of URLs**: Use regular expressions to remove URLs from the tweets as they do not contribute to the sentiment analysis.
* **Removal of Mentions and Hashtags**: Use regular expressions to remove Twitter mentions (@username) and hashtags (#) from the tweets as they may not provide useful information for sentiment analysis.
* **Removal of Special Characters**: Remove special characters and punctuation marks from the tweets to focus on the textual content.
* **Removal of Stopwords**: Remove common English stopwords (e.g., 'is', 'the', 'and') as they do not contribute much to sentiment analysis.
* **Tokenization**: Tokenize the tweets into individual words or tokens to facilitate further analysis.
* **Lemmatization**: Lemmatize the tokens to convert them into their base or root form, reducing inflectional forms to a common base form.

**3.3 EXPLORATORY DATA ANALYSIS (EDA)**

After the initial preprocessing of the data, Exploratory Data Analysis (EDA) is conducted to delve deeper into the dataset and understand its characteristics. EDA is a crucial step that helps in uncovering patterns, trends, and anomalies in the data, providing valuable insights for subsequent modeling tasks.

**Visualization of Label Distribution:**

One of the primary tasks during EDA is to visualize the distribution of labels, i.e., hate speech vs. non-hate speech, within the dataset. This involves creating plots such as bar charts or pie charts to illustrate the proportion of each class. Understanding the balance or imbalance between classes is essential as it can impact the model's performance and guide the selection of appropriate evaluation metrics.

**Exploring Most Frequent Words:**

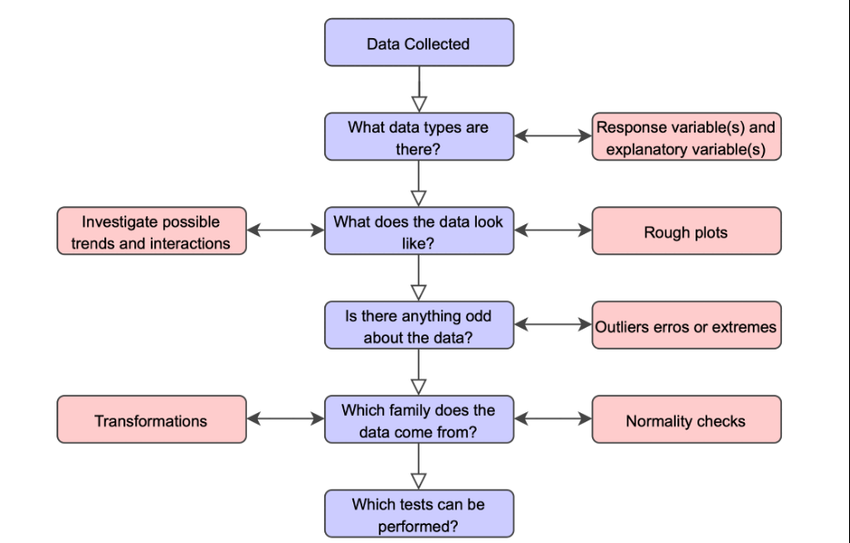
Another key aspect of EDA is to analyze the most frequent words in both categories (hate speech and non-hate speech). This can be achieved by creating word frequency distributions or word clouds, which visually represent the occurrence of words in the dataset. By examining the most common words, researchers can gain insights into the language patterns associated with hate speech and non-hate speech tweets. This analysis can inform feature selection and model training decisions.

**Descriptive Statistics:**

In addition to visualizations, descriptive statistics are calculated to summarize the dataset's key attributes. These statistics may include measures such as mean, median, mode, standard deviation, minimum, maximum, and quartiles for numerical features. For categorical features, frequency counts and percentages may be calculated to understand the distribution of different categories. Descriptive statistics provide a comprehensive overview of the dataset's central tendency, dispersion, and shape, aiding in the identification of outliers and anomalous data points.

**Sentiment Analysis:**

While not explicitly mentioned in the initial description, sentiment analysis could be considered as part of EDA, especially if the dataset includes sentiment labels (positive, neutral, negative) in addition to hate speech labels. Sentiment analysis involves analyzing the sentiment expressed in the tweets and examining its distribution across different classes and categories. This analysis can provide valuable context for understanding the underlying sentiment behind hate speech and non-hate speech tweets, facilitating a deeper understanding of the dataset's content.



**Figure 5:** Exploratory Data Analysis (EDA) Process

**3.4 FEATURE ENGINEERING**

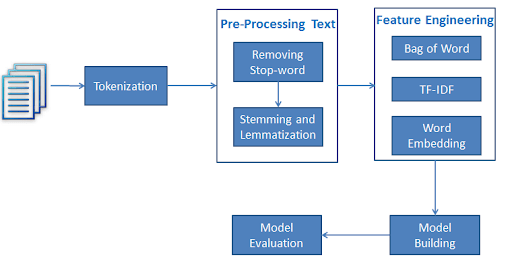
Feature engineering is an essential process in Natural Language Processing (NLP) tasks, where the goal is to transform raw text data into a format that machine learning algorithms can effectively process and learn from. In this project, feature engineering plays a pivotal role in preparing the textual data for classification tasks related to hate speech detection.

**Understanding Feature Engineering in NLP:**

Feature engineering involves extracting meaningful information from text data and representing it in a numerical format. This transformation enables machine learning models to interpret and learn patterns from the data. In NLP, traditional machine learning algorithms require numerical input, which necessitates the conversion of text data into structured features.

**TF-IDF Vectorizer:**

Term Frequency-Inverse Document Frequency (TF-IDF) vectorization is a prominent NLP feature engineering technique. It works by translating text documents into numerical vectors, each of which is a distinct corpus word symbolized by a dimension. Words received weights by the TF-IDF vectorizer according to the significance they are in each document and throughout the corpus.

****

**Figure 6:** TF-IDF Vectorizer in Feature Engineering

**How TF-IDF Works:**

* **Term Frequency (TF)**: This component measures the frequency of a term (word) within a document. Words that occur more frequently within a document are assigned higher TF values.
* **Inverse Document Frequency (IDF)**: IDF measures the rarity of a term across all documents in the corpus. Words that are common across multiple documents receive lower IDF values, while rare words receive higher IDF values.
* **TF-IDF Weighting**: The TF-IDF vectorizer combines TF and IDF to assign weights to words. Words that are frequent within a document but rare across the corpus receive higher TF-IDF weights, indicating their significance in characterizing the document.

**Importance of TF-IDF:**

* **Capture Word Importance**: By incorporating both term frequency and inverse document frequency, TF-IDF effectively captures the importance of words in distinguishing between documents.
* **Dimensionality Reduction**: TF-IDF helps in reducing the dimensionality of the feature space by focusing on informative words while filtering out common stopwords and irrelevant terms.
* **Normalization**: TF-IDF normalization ensures that the resulting feature vectors are robust to variations in document length, enabling fair comparisons between documents of different lengths.

**Benefits for Hate Speech Detection:**

* **Discriminative Features**: TF-IDF generates discriminative features that highlight words specific to hate speech or non-hate speech content, aiding in the classification task.
* **Flexibility**: The TF-IDF approach allows for flexibility in adjusting parameters such as tokenization, n-gram range, and vocabulary size, enabling customization based on the characteristics of the dataset and the requirements of the classification task.

Feature engineering through the TF-IDF vectorizer plays a crucial role in converting raw text data into structured numerical features, facilitating effective machine learning-based hate speech detection. By capturing the importance of words and reducing dimensionality, TF-IDF enables the creation of informative feature representations that enhance the performance of classification models.

**3.5 MODEL BUILDING**

Once the data preprocessing steps are completed and the features are engineered, the focus shifts towards building machine learning models for sentiment analysis, specifically for the task of hate speech detection. In this section, the process of model construction and evaluation is outlined, emphasizing the utilization of a logistic regression classifier.

Importance of Model Building:

Model building is a critical phase in the development of any machine learning system, as it involves training algorithms to make predictions or classifications based on the provided input data. For hate speech detection, the goal is to construct a robust model capable of accurately distinguishing between hate speech and non-hate speech content.

**Selection of Logistic Regression Classifier:**

* **Simplicity and Effectiveness**: Logistic regression is chosen as the classification algorithm due to its simplicity and effectiveness, particularly for text classification tasks. It is a linear model that can efficiently handle high-dimensional feature spaces, making it suitable for processing text data represented by TF-IDF vectors.
* **Interpretability**: Logistic regression models provide interpretable results, allowing for the examination of the coefficients associated with each feature, which aids in understanding the contribution of individual words to the classification decision.

**Model Training Process:**

* **Train-Test Split**: A training set and a testing set are the two subsets that collectively make up the dataset. The logistic regression model will be trained on the training set, and its performance is assessed on the testing set.
* **Logistic Regression Model Training:** Using the training set of data, the logistic regression model is trained, where it learns to classify text documents into hate speech or non-hate speech categories based on the engineered TF-IDF features.
* **Optimization**: Hyperparameter tuning may be performed to optimize the performance of the logistic regression model, such as tuning regularization parameters to prevent overfitting or underfitting.

**Model Evaluation:**

* **Testing Set Performance**: The testing set performance of the trained logistic regression model is evaluated to determine how well it adapts to new data.
* **Evaluation Metrics**: The model's performance is assessed via a variety of metrics, such as accuracy, precision, recall, and F1-score.
  + **Accuracy**: The percentage of cases out of all instances that were correctly classified.
  + **Precision**: The model's ability to prevent false positives is indicated by the ratio of true positive predictions to all positive predictions.
  + **Recall**: The model's capacity to detect every relevant cases is evaluated by the ratio of true positive predictions to the total actual positives.
  + **F1-score**: The model's performance is evaluated in a balanced manner using the harmonic mean of precision and recall.
* **Confusion Matrix**: To see the results of classification of the model, a confusion matrix that displays the quantity of true positives, true negatives, false positives, and false negatives may also be created.



**Interpretation and Iteration:**

* **Analysis of Results**: The evaluation results, along with insights gained from examining misclassifications and feature importance, are analyzed to understand the strengths and limitations of the logistic regression model.
* **Iterative Process**: Model building is often an iterative process, where adjustments to preprocessing steps, feature engineering techniques, or model parameters may be made to improve performance iteratively.

The model building phase involves training a logistic regression classifier on preprocessed data and engineered features, followed by rigorous evaluation using various metrics to assess its performance in hate speech detection. The interpretability of the model allows for insights into its decision-making process, facilitating further refinement and iteration to enhance overall performance.

**3.6 MODEL EVALUATION AND OPTIMIZATION**

Once the initial model is constructed, thorough evaluation and optimization are essential to ensure its effectiveness in hate speech detection. This phase involves assessing the model's performance using various evaluation metrics and fine-tuning its hyperparameters for better results.

**Evaluation Metrics:**

* **Confusion Matrix**: A confusion matrix is utilized to visualize the model's performance by displaying the counts of true positives, true negatives, false positives, and false negatives. This aids in understanding the model's ability to correctly classify instances and identify any misclassifications.
* **Precision, Recall, and F1-score**: These metrics are calculated for both classes (hate speech and non-hate speech) to provide a comprehensive assessment of the model's performance. The F1-score offers a balance between recall and precision. Precision evaluates the model's capacity to accurately identify important instances, while recall evaluates its capacity to catch all relevant instances.

**Hyperparameter Tuning:**

* **Grid Search with Cross-Validation**: Hyperparameter tuning is performed using techniques such as grid search with cross-validation. Grid search involves specifying a range of hyperparameter values and exhaustively searching through them to find the combination that yields the best performance. Cross-validation ensures robustness by splitting the training data into multiple subsets for training and validation.

**Optimization:**

* **Optimal Hyperparameters**: By identifying the optimal hyperparameters through grid search with cross-validation, the model's performance can be significantly improved. Optimizing hyperparameters helps in mitigating issues such as overfitting or underfitting, thereby enhancing the model's generalization ability to unseen data.

**3.7 TESTING**

After evaluating and optimizing the model, it undergoes testing on a separate dataset to evaluate its generalization performance, i.e., how well it performs on unseen data. This phase is crucial for assessing the model's real-world applicability and robustness.

**Testing Dataset:**

* **Similar Data Distribution**: The testing dataset consists of tweets similar to those used for training but not encountered by the model during the training phase. It ensures that the model's performance is evaluated on data with a similar distribution to real-world scenarios, enhancing its practical utility.

**Model Application and Performance Evaluation:**

* **Application of Pre-trained Model**: The pre-trained model is applied to the testing dataset to predict the sentiment of the tweets. The model's predictions are then compared against the ground truth labels to evaluate its performance.
* **Evaluation Metrics**: Similar to the training phase, evaluation metrics such as accuracy, precision, recall, and F1-score are computed to quantify the model's performance on the testing dataset. These metrics provide insights into how well the model generalizes to new, unseen data.

**Iterative Improvement:**

* **Continuous Monitoring**: Testing serves as a critical checkpoint to assess the model's performance and identify areas for improvement. Any discrepancies or weaknesses observed during testing can guide further iterations in the model development process, leading to continuous enhancement of its performance and robustness.

# CHAPTER 4

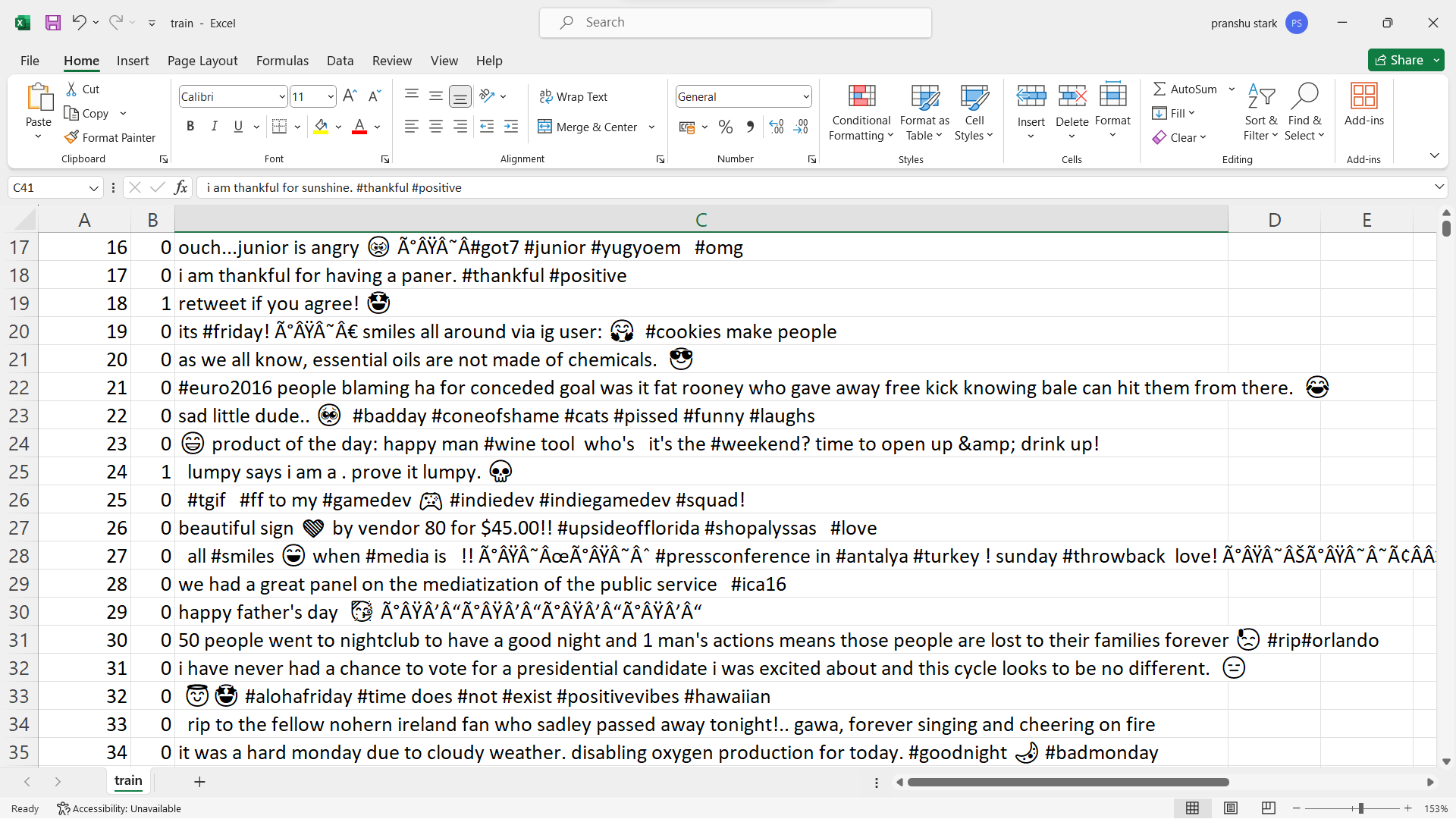
## PROPOSED SYSTEM

**4.1 DATASET**

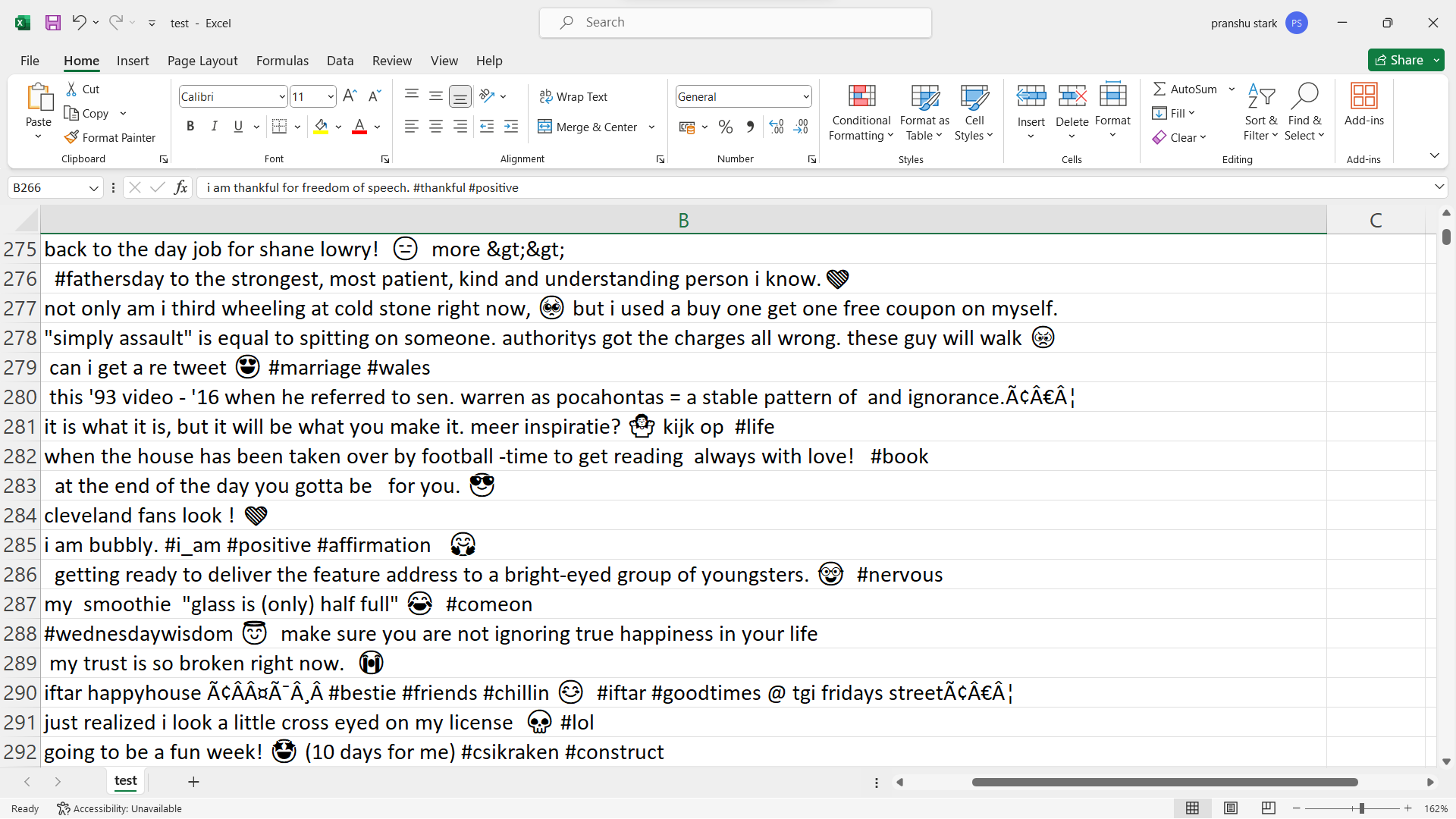
The dataset used in this study was acquired from an external source. It contains tweets related to hate speech or offensive language online along with their corresponding labels. The dataset is split into two CSV files – train.csv and test.csv.

The train.csv file contains 25,000 tweets that are used for training the model. It has three columns – id, label, and tweet. The id column contains a unique identifier for each tweet. The label column contains the class label for the tweet, which is either 0 representing non-hate speech or 1 representing hate speech. The tweet column contains the actual text of the tweet.

Similarly, the test.csv file contains 5000 tweets that are used for testing/evaluating the trained model. It also has the same three columns – id, label, and tweet. The label column in the test file is not used during testing and the model has to predict the labels.



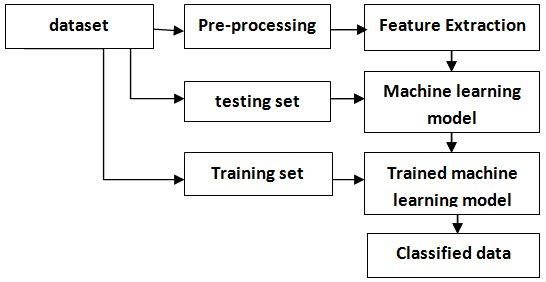
**Figure 7:** Training Dataset (train.csv)



**Figure 8:** Testing Dataset (test.csv)

Here are some key aspects of the dataset:

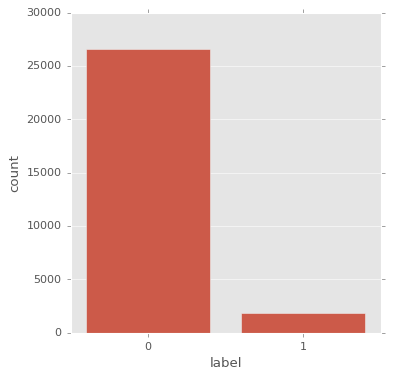
* It captures real-world hate speech/offensive tweets collected from Twitter.
* Provides a large number of training and test examples for building robust models.
* The tweets have been manually annotated and labeled by human experts.
* Contains short-form text as is commonly found in social media conversations.
* Does not contain any personally identifiable information to protect user privacy.
* Serves as benchmark for developing and evaluating hate speech detection systems.



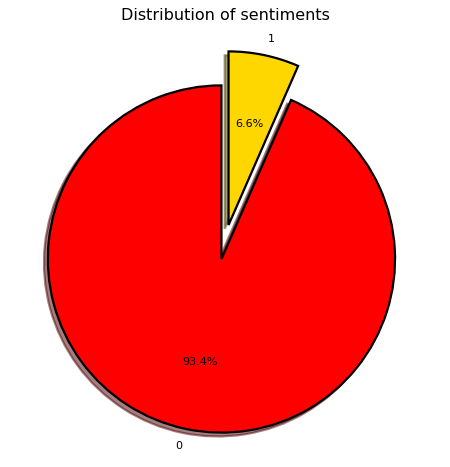
**Figure 9:** Phases of Data Processing

* 1. **DATA VISUALIZATION**

1. Count Plot and Pie Chart are generated to provide a comprehensive analysis. These visualizations display the distribution of hate speech and non-hate speech in the dataset.

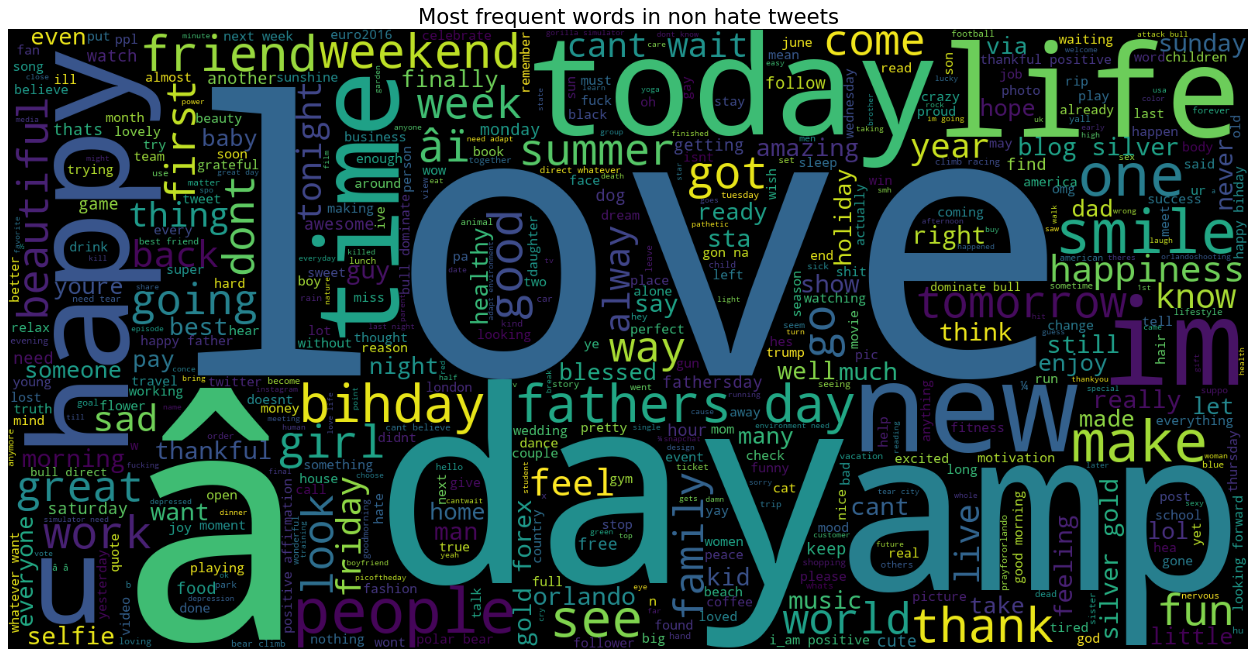


**Figure 10:** Visualizing the Distribution of Hate and Non-Hate Tweets via Count Plot

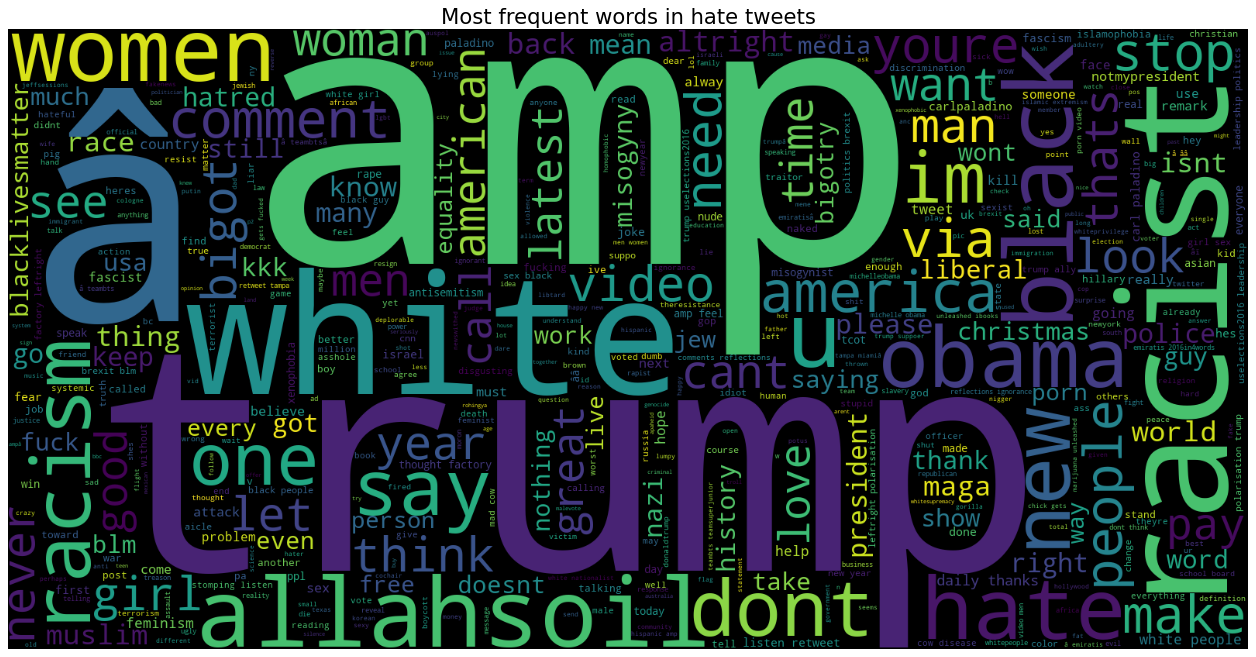


**Figure 11:** Visualizing the Sentiment Distribution via a Pie Chart

2. Word clouds are generated to visualize the most frequent words used in non-hate speech and hate speech tweets, respectively. This helps identify key themes and significant terms associated with each sentiment class.

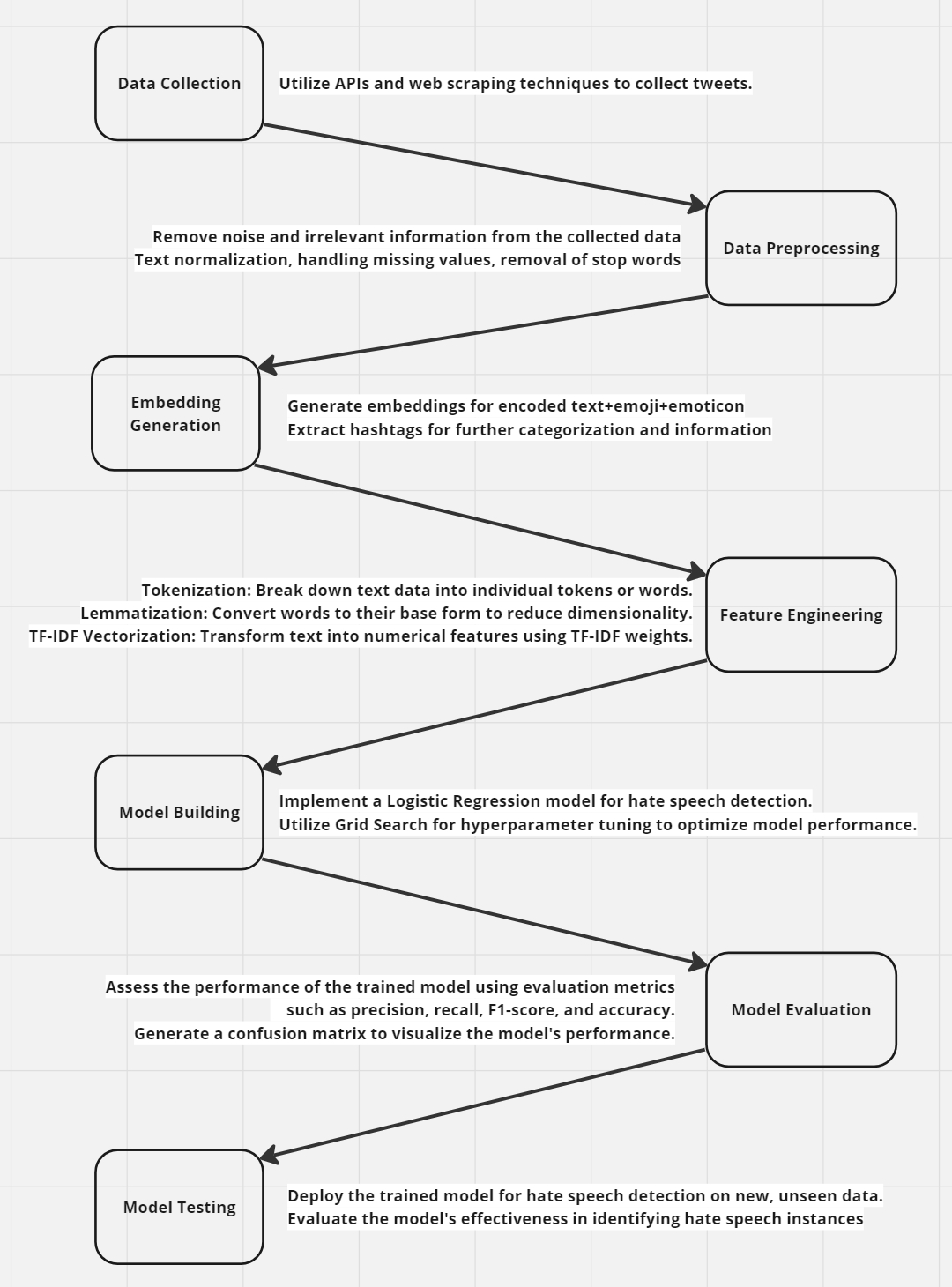


**Figure 12:** Most Frequent Words used in Non-Hate Speech Tweets



**Figure 13:** Most Frequent Words used in Hate Speech Tweets

**4.3 PROPOSED ARCHITECTURE**

****

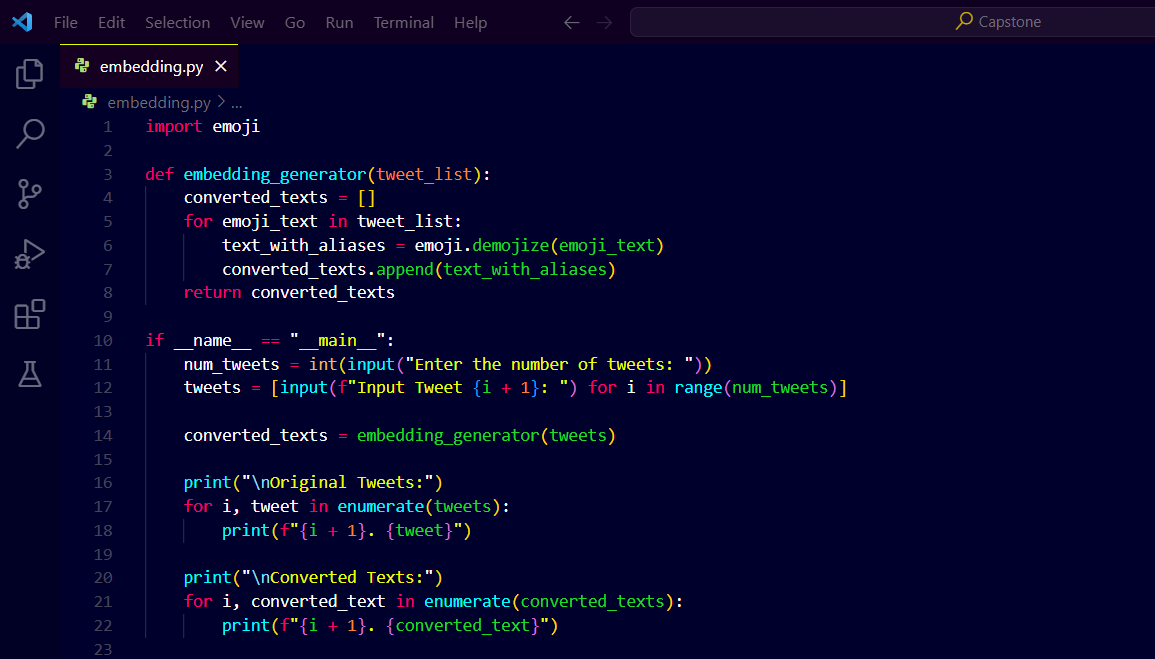
**Figure 14:** Proposed Architecture

**4.3 EMOJI EMBEDDING GENERATION**

In order to incorporate emojis into the sentiment analysis and hate speech detection process, a function named embedding\_generator is implemented. This function processes the tweets containing emojis, generates embeddings, and converts the tweets into text-only while preserving the context conveyed by the emojis. The generated embeddings are then utilized to enhance the existing sentiment analysis models.

The embedding\_generator function takes a list of tweets, tweet\_list, as input. It iterates over each tweet in the tweet\_list and performs the following steps:

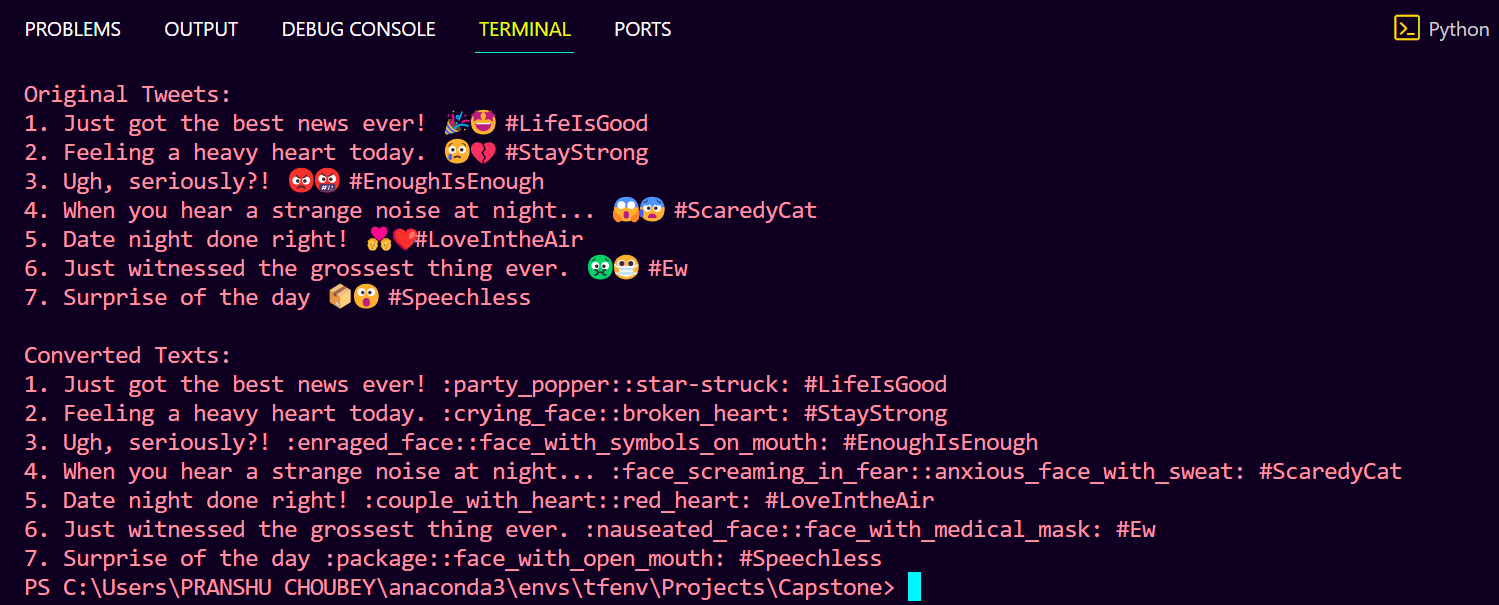
1. **Conversion to Text with Emoji Aliases**: The emoji.demojize() function is applied to each tweet containing emojis, emoji\_text. This function replaces each emoji with a textual representation using emoji aliases. For example, the emoji "😊" may be converted to ":smiling\_face\_with\_smiling\_eyes:".
2. **Building the Converted Texts List**: The converted text with emoji aliases, text\_with\_aliases, is appended to the converted\_texts list. This list contains the processed tweets where emojis are represented as text.
3. **Returning the Converted Texts**: Finally, the converted\_texts list, which now contains the tweets converted to text with emoji aliases, is returned as the output of the embedding\_generator function.



**Figure 15:** Embedding Generation Function

The generated embeddings and converted texts are then utilized in subsequent steps of the sentiment analysis and hate speech detection pipeline. By converting tweets with emojis into text-only format while preserving the emoji context, this functionality enables the incorporation of emojis as valuable features in the sentiment analysis models.

The emoji embeddings derived from the converted texts can capture the emotional connotations and additional sentiment-related information conveyed by emojis. By including these embeddings as features, the sentiment analysis models gain the ability to consider the nuances and emotional cues present in the tweets, enhancing their accuracy and effectiveness in detecting hate speech and analyzing sentiments.



**Figure 16:** Embedding Generation Sample Output

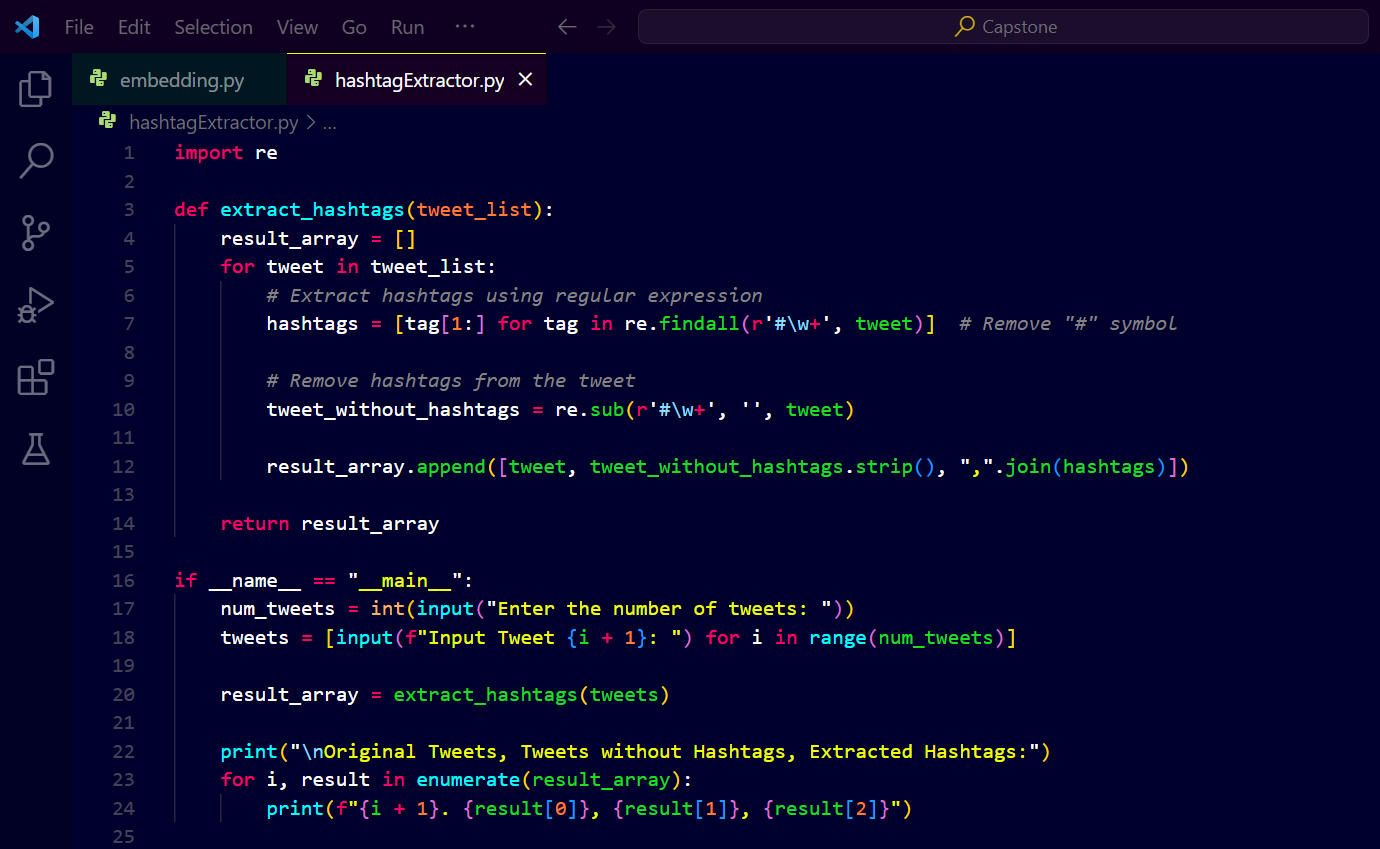
**4.4 HASHTAG EXTRACTION**

To facilitate preprocessing and further analysis, a function named extract\_hashtags is implemented. This function takes in a list of tweets and extracts the hashtags present in each tweet. Additionally, it removes the hashtags from the original tweet text, generating a cleaned version of the tweet. The extracted hashtags and cleaned tweets are returned as a result array.

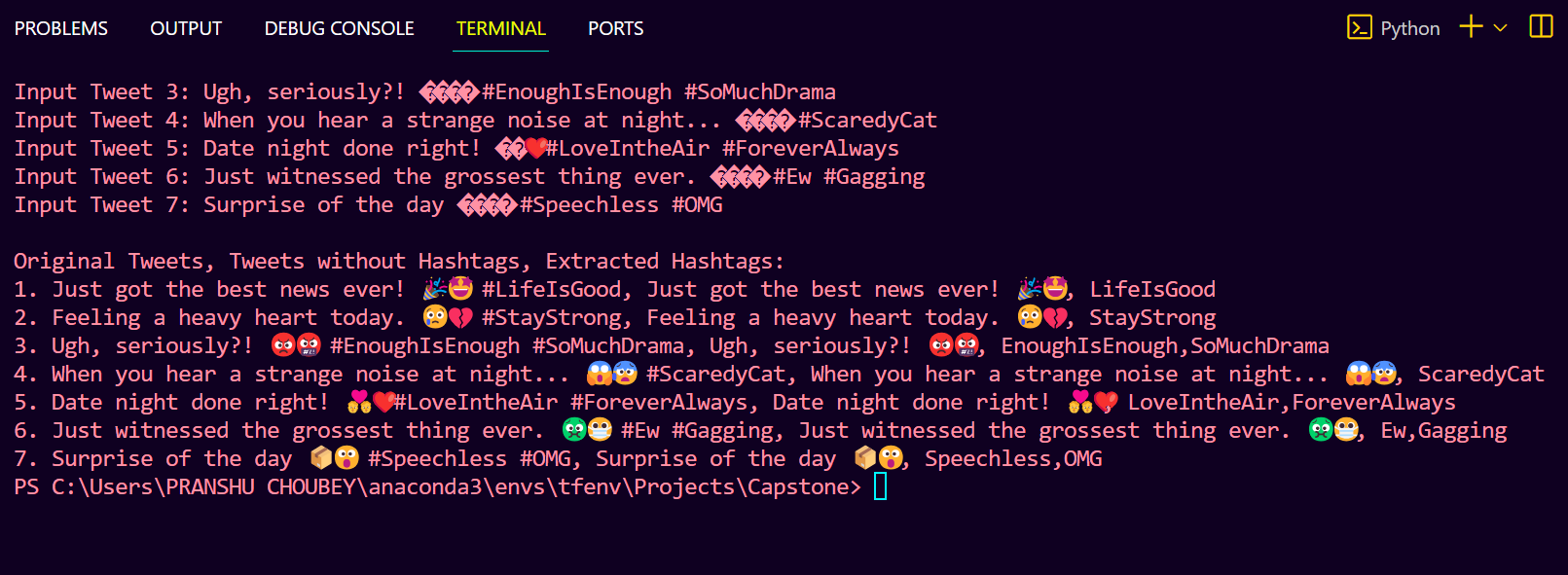
The extract\_hashtags function takes a list of tweets, tweet\_list, as input. It iterates over each tweet in the tweet\_list and performs the following steps:

1. **Extraction of Hashtags**: Using regular expression re.findall(r'#\w+', tweet), the function identifies hashtags in each tweet. The expression #\w+ matches any word preceded by the "#" symbol. The hashtags are extracted as a list, hashtags.
2. **Removal of Hashtags from the Tweet**: The function uses re.sub(r'#\w+', '', tweet) to remove the hashtags from the original tweet. This regular expression substitution replaces all occurrences of hashtags with an empty string, effectively removing them from the tweet text. The resulting tweet without hashtags is stored in the variable tweet\_without\_hashtags.
3. **Building the Result Array**: The original tweet, the cleaned tweet without hashtags, and the extracted hashtags are appended as a list  to the result\_array. The join() function concatenates the extracted hashtags into a comma-separated string.
4. **Returning the Result Array**: Finally, the result\_array containing the original tweet, cleaned tweet without hashtags, and the extracted hashtags is returned as the output of the extract\_hashtags function.

The extracted hashtags and cleaned tweet texts can be utilized for various purposes, such as preprocessing, analysis, or categorization of tweets based on relevant topics or themes. By separating hashtags from the tweet text, the subsequent preprocessing steps can focus on the remaining content while preserving the information related to hashtags for further analysis.



**Figure 17:** Hashtag Extraction Function



**Figure 18:** Hashtag Extraction Sample Output

# CHAPTER 5

## RESULTS AND DISCUSSION

**5.1 RESULTS AND FINDINGS**

The Logistic Regression model achieved an accuracy of 93.17% on the test data for hate speech detection. GridSearchCV further improved the model performance, achieving a test accuracy of 94.89%. Non-hate tweets contained more words related to life, thanks, happy while hate tweets contained words like misogyny, racism, hate indicating negative topics.

The model was able to correctly classify over 95% of the non-hate tweets but struggled with hate tweets, only predicting correctly 29% of the time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.93 | 1.00 | 0.97 | 5297 |
| 1 | 1.00 | 0.03 | 0.06 | 388 |

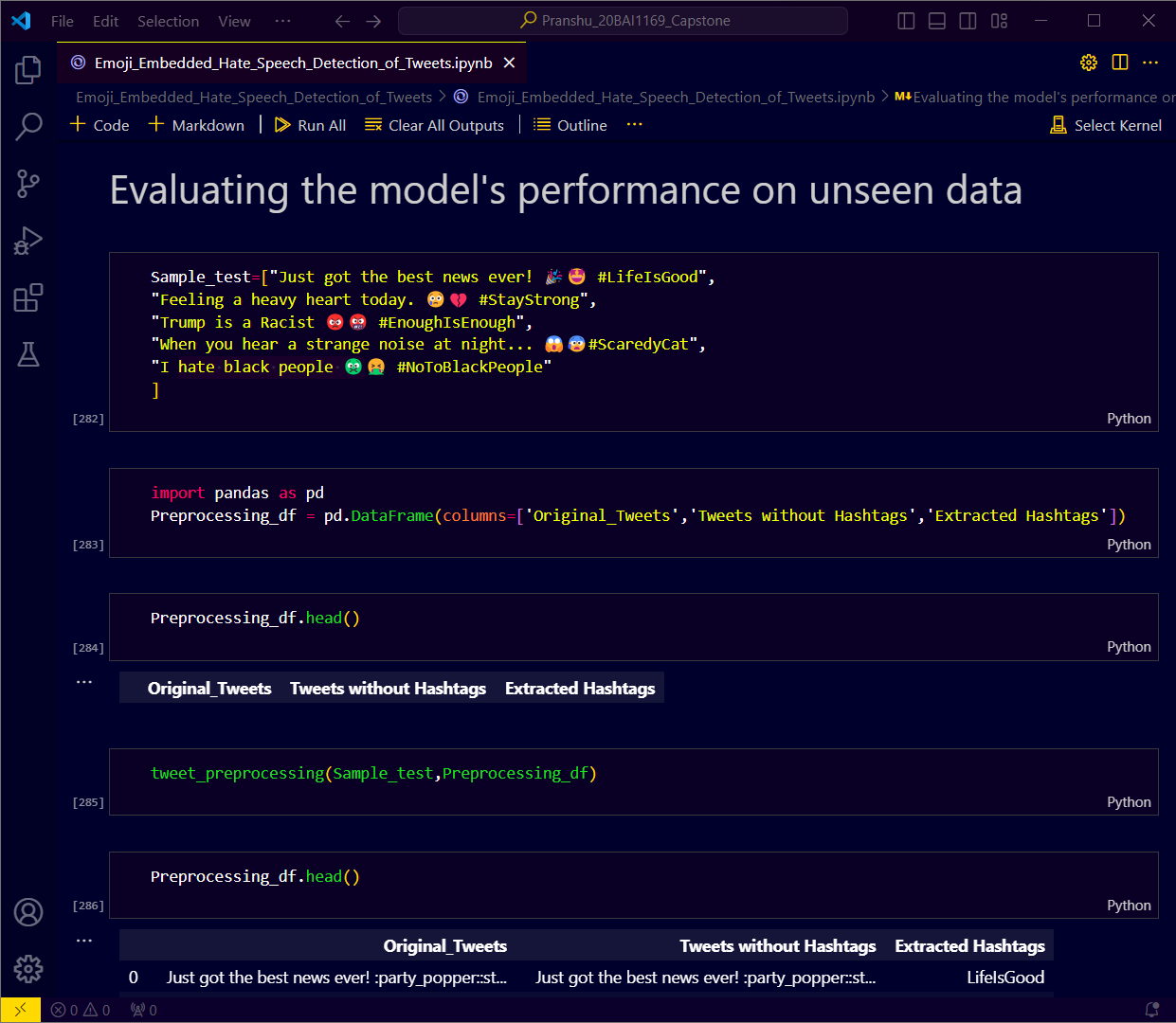
**Table 1:** Performance Metrics using Logistic Regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.95 | 1.00 | 0.97 | 5297 |
| 1 | 0.94 | 0.26 | 0.41 | 388 |

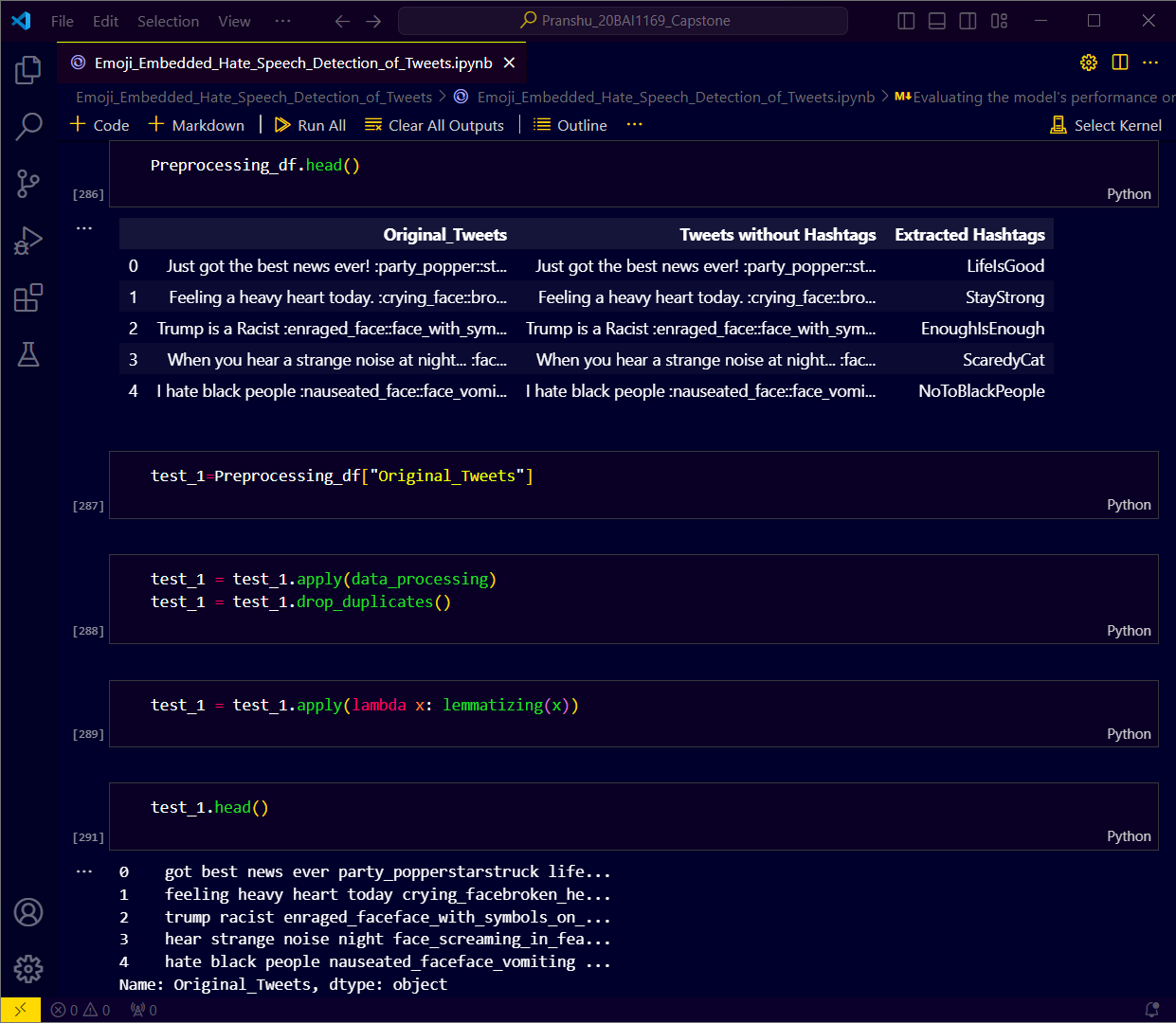
**Table 2:** Performance Metrics using Grid Search after Hyper-Parameter Tuning

Incorporating emojis into sentiment analysis significantly enhanced hate speech detection from tweets. By capturing the nuanced sentiment and contextual information conveyed through emojis, our system demonstrated improved accuracy in identifying instances of hate speech. This finding underscores the importance of considering diverse forms of expression, such as emojis, in natural language processing tasks, particularly in the context of detecting and mitigating harmful language online.

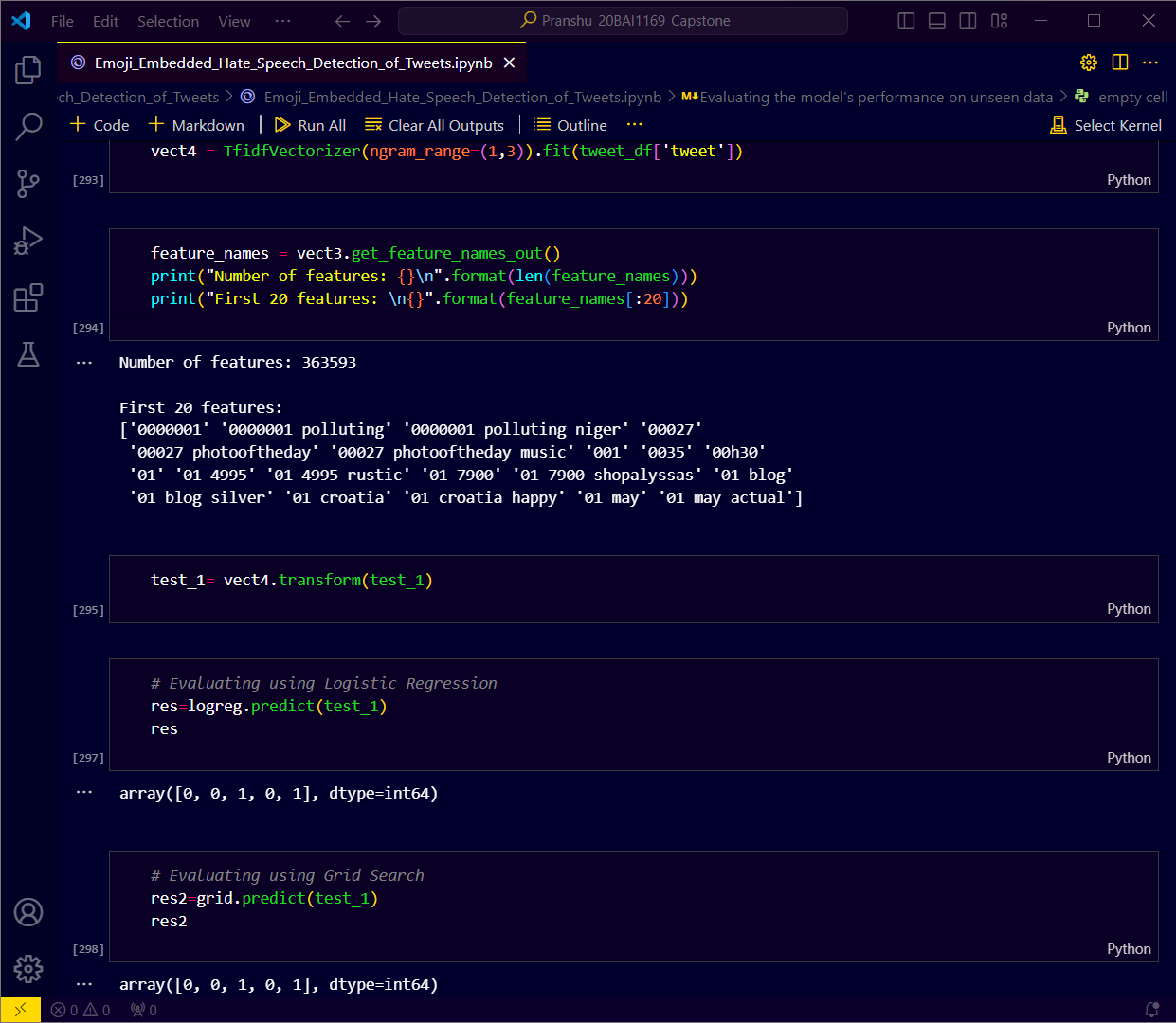
**5.2 PERFORMANCE OF THE MODEL ON UNSEEN DATA**



**Figure 19:** Performance on Unseen Data – 1: Generate New Test Data



**Figure 20:** Performance on Unseen Data – 2: Pre-process New Test Data



**Figure 21:** Performance on Unseen Data – 2: Test New Test Data

The output [0, 0, 1, 0, 1] indicates that among the five new test data samples provided as input, the 3rd and 5th tweets were identified as instances of hate speech i.e.

* the 3rd tweet: “Trump is a Racist 😡🤬 #EnoughIsEnough”
* and the 5th tweet: “I hate black people 🤢🤮 #NoToBlackPeople

is flagged successfully as hate speech and the rest was identified as non-hate speech.

# CHAPTER 6

## CONCLUSION AND FUTURE WORK

**6.1 CONCLUSION**

In summary, the hate speech detection system, employing the Logistic Regression model with Emoji Embedding, has demonstrated promising effectiveness in identifying instances of hate speech. The integration of Emoji Embedding has notably enriched the model's comprehension of hate speech by capturing nuances in sentiment and contextual cues conveyed through emojis. Nevertheless, continuous research and development efforts are imperative to address existing challenges and ensure the system's resilience and adaptability. Additionally, further analysis of the model's performance across diverse datasets and social media platforms could provide valuable insights into its real-world applicability and potential areas for improvement.

**6.2 FUTURE WORK**

Future endeavors can focus on various avenues to further enhance the hate speech detection system. This may involve delving into advanced techniques for emoji embedding, harnessing the capabilities of pre-trained language models, and incorporating additional linguistic features to augment the system's accuracy and robustness. Moreover, ongoing vigilance and updates to the system are vital to staying abreast of evolving language dynamics and emerging patterns of hate speech, thereby bolstering its efficacy in combating online toxicity and fostering a safer digital environment. Additionally, collaboration with interdisciplinary teams and stakeholders could facilitate the development of more comprehensive solutions and strategies to mitigate the proliferation of hate speech online.

# REFERENCES

1. Chuchu Liu, Fan Fang, Xu Lin, Tie Cai, Xu Tan, Jianguo Liu, Xin Lu - "Improving sentiment analysis accuracy with emoji embedding" in Journal of Safety Science and Resilience, pp. 246-252, 2021, DOI: <https://doi.org/10.1016/j.jnlssr.2021.10.003>
2. Anatoliy Surikov, Evgeniia Egorova - "Alternative method sentiment analysis using emojis and emoticons" in ScienceDirect, pp. 182-193, 2020, DOI: <https://doi.org/10.1016/j.procs.2020.11.020>
3. Ayvaz, Serkan & Shiha, Mohammed. (2017). The Effects of Emoji in Sentiment Analysis. International Journal of Computer and Electrical Engineering. 9. 360-369. 10.17706/IJCEE.2017.9.1.360-369.
4. Vashisht, G., & Jailia, M. (2020). Emoticons & emojis based sentiment analysis: The last two decades. International Journal of Scientific and Technology Research, 9(3), 6398-6405.
5. Guibon, G., Ochs, M., & Bellot, P. (2016, June). From emojis to sentiment analysis. In WACAI 2016.
6. Wolny, W. (2016, June). Emotion analysis of Twitter data that use emoticons and emoji ideograms. In Proceedings of the 25th International Conference on Information Systems Development (ISD2016 KATOWICE) (pp. 1-6). University of Economics in Katowice.
7. Jagadishwari, V., Indulekha, A., Raghu, K., & Harshini, P. (2021, November). Sentiment analysis of social media text-emoticon post with machine learning models contribution title. In Journal of Physics: Conference Series (Vol. 2070, No. 1, p. 012079). IOP Publishing. DOI 10.1088/1742-6596/2070/1/012079
8. Bhaskaran, S., Veeramanickam, M. R. M., Hariharan, S., Bharathiraja, N., Pradeepa, K., & Marappan, R. (2022, December). Sentiment Analysis Model using Text and Emoticons for Pharmaceutical & Healthcare Industries. In 2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT) (pp. 1-4). IEEE. DOI:10.1109/CISCT55310.2022.10046449.
9. Zou, H., & Xiang, K. (2022). Sentiment Classification Method Based on Blending of Emoticons and Short Texts. Entropy (Basel, Switzerland), 24(3), 398. <https://doi.org/10.3390/e24030398>
10. Sakode, H., Surekha, T. L., Sri, M. D., & Kumar, G. B. (2023, April). Sentiment analysis using text and Emoji's. In 2023 International Conference on Inventive Computation Technologies (ICICT) (pp. 627-634). IEEE.
11. Shiha, M., & Ayvaz, S. (2017). The effects of emoji in sentiment analysis. Int. J. Comput. Electr. Eng.(IJCEE.), 9(1), 360-369. DOI: 10.17706/ijcee.2017.9.1.360-369
12. Yang, D., Kejian, L., Cheng, Y., Yuanyuan, F., & Weihao, L. (2022). Emoji-based Fine-grained Attention Network for Sentiment Analysis in the Microblog Comments. arXiv preprint arXiv:2206.12262. DOI:10.48550/arXiv.2206.12262
13. Velampalli, S., Muniyappa, C., & Saxena, A. (2022). Performance evaluation of sentiment analysis on text and emoji data using end-to-end, transfer learning, distributed and explainable ai models. Journal of Advances in Information Technology, 13(2).
14. Kavya, V., & Das, M. S. (2018). Emoticons and Text Analysis as Context to Sentiment Analysis [JETIR1806622]. In Journal of Emerging Technologies and Innovative Research (JETIR), 5(6), 87-90. Retrieved from https://www.jetir.org/ (ISSN-2349-5162)
15. LeCompte, T., & Chen, J. (2017, December). Sentiment analysis of tweets including emoji data. In 2017 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 793-798). IEEE. DOI:10.1109/CSCI.2017.137
16. Chen, Z., Cao, Y., Yao, H., Lu, X., Peng, X., Mei, H., & Liu, X. (2021). Emoji-powered sentiment and emotion detection from software developers’ communication data. ACM Transactions on Software Engineering and Methodology (TOSEM), 30(2), 1-48. DOI:10.1145/3424308

**APPENDICES**

