



Implementation of Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence

by

Student Name

Department / Program Name

College Name

University of Sharjah

Sharjah, UAE

Supervisor(s)

Main Supervisor: [insert name]

Co-Supervisor: [insert name if exists. Delete if it does not exist]

A Dissertation Submitted in Partial Fulfilment of the Requirements for [insert
the program name]

University of Sharjah
Sharjah, UAE

Date: [Insert current Date]

The examination committee approves the dissertation/thesis of the Student's Name.

[Insert the name of examiner 1].....Committee Chair
[Insert the title and affiliation of Examiner 1]

[Insert the name of examiner 2].....member
[Insert the title and affiliation of Examiner 2]

[Insert the name of examiner 3].....member
[Insert the title and affiliation of Examiner 3]

[Insert the name of examiner four if exists, delete if it does not exist].....member
[Insert the title and affiliation of Examiner 4]

© Approval Date

(Date must be the same as the year that the author received his/her thesis approval)

Student Name

All Rights Reserved

Abstract

Implementation of Novel Opinion Mining Approach on Digital Media Content using
Artificial Intelligence

Abdulrahman Radaideh

The rapid evolution of digital media platforms has spurred the proliferation of user-generated content, underscoring the need for robust methodologies in deciphering sentiments embedded within this content. This thesis embarks on an exploration of implementing a groundbreaking Opinion Mining Approach utilizing Artificial Intelligence (AI) to dissect sentiments within digital media, specifically focusing on ChatGPT Sentiment Analysis.

The research objectives were meticulously designed to unravel the significance of sentiment analysis in social media platforms, synthesize existing literature, and propose an innovative framework for predicting sentiments from user reviews. The study unfolds in addressing pivotal research questions, examining the importance of sentiment analysis, surveying data collection sources and methodologies, and devising strategies for converting diverse user reviews - from video, audio, and image formats into text for sentiment analysis.

The research culminated in an inventive framework involving sentiment extraction from multi-modal data, clustering sentiments into positive, negative, and neutral categories, and performing in-depth sentiment labeling of tweets. Objective assessments and detailed analyses illuminated the critical role of sentiment analysis in informing decision-making processes and enhancing quality assurance within digital media landscapes.

The findings of this study bear significant implications for decision-making processes, as the novel AI-driven approach enriches understanding of user sentiments, guiding more informed decisions. Additionally, the study's enhanced sentiment analysis techniques offer opportunities for improved content curation, user satisfaction, and quality assurance across diverse social media platforms.

While the research has made substantial strides, certain limitations pertaining to data collection methodologies, model accuracy, and interpretational nuances were encountered. Addressing these limitations presents avenues for future research to refine methodologies and bolster the robustness of AI-driven Opinion Mining.

In conclusion, the Implementation of Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence signifies a crucial step towards unraveling user sentiments in the digital sphere. As digital landscapes continue to evolve, the research underscores the pivotal role of AI in deciphering sentiments and enhancing decision-making processes across multifaceted digital media platforms.

ملخص (باللغة العربية) (Mandatory)

تحتوي هذه الصفحة على ترجمة للملخص باللغة العربية. يرجاء استخدام لغة عربية فصحى وعدم استخدام البرامج او المواقع الالكترونية فى الترجمة. ينصح بالاستعانة بمتخصص حتى تكون اللغة واضحة ومفهومة!

Time new Roman, 12 font sizes, and double space lines

Author's Biographical Sketch (**Mandatory**)

Times New Roman font of size 12. The line space is set to 2.0.

Insert Biographical Sketch text here. If you wish to use or paste text from another document, select it from the other document and paste it here. Make sure to keep the text-only and not the formatting from the previous document. To mark text only, choose Paste, and then from the drop-down box that will appear, select the Keep Text Only option on the right, with the icon of a clipboard and the letter A.

This is the beginning of the second paragraph.

Declaration (**Mandatory**)

Times New Roman font of size 12. The line space is set to 2.0

Insert Dedication text here. If you wish to use or paste text from another document, select it from the other document and paste it here. Make sure to keep the text-only and not the formatting from the previous document. To keep text only, choose Paste, and then from the drop-down box that will appear, choose the Keep Text Only option on the right, with the icon of a clipboard and the letter A.

This is the beginning of the second paragraph.

Example for declaration:

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification and that it is the result of my own independent work.

I also declare that there is no conflict of interest.

Any figures or tables that have been published before should be reproduced or get copyright.

I declare that based on the similarity check the total similarity is not more than 15%, excluding the bibliography (references) and quotes.

Full Name Goes Here (Candidate)

Date

Acknowledgments (**Mandatory**)

Times New Roman font of size 12. The line space is set to 2.0Insert

Acknowledgments text here. If you wish to use or paste text from another document, select it from the other document and paste it here. Make sure to keep the text-only and not the formatting from the previous document. To keep text only, choose Paste, and then from the drop-down box that will appear, choose the Keep Text Only option on the right, with the icon of a clipboard and the letter A.

This is the beginning of the second paragraph.

The Acknowledgements page is mandatory. It follows the Abstract page. It is in the same font size and style as the text, and the vertical spacing, paragraph style margins, and alignment are the same as used in the text. Use complete sentences.

(Sample Wording)

I would like to thank my committee chair, Dr. Mohammad...., and my committee members, Dr. Ahmed, Dr. Hamad, Dr. Rashed, and Professor Sultan, for their guidance and support throughout the course of this research.

My appreciation also goes to my friends and colleagues and the department faculty and staff for making my time at the University of Sharjah a great experience. I also want to extend my gratitude to the RISE, RIMHS which provided the survey instrument, and to all the Department of..... Professors and students who were willing to participate in the study.

Finally, my heartfelt gratitude is extended to my parents for their encouragement and my wife's patience and support.

In addition to the above statement please include the following:

- **Acknowledge all funding sources, including funds from College of Graduate Studies, projects titles and numbers (if applicable)**

- **Ethical approval (if applicable).**

In case you conducted research that involves animals or humans, you should include the approval from the concerned university committee.

Publications associated with this research (Mandatory)

Paper 1

A Novel Approach to Predict the Real Time Sentimental Analysis by Naive Bayes & RNN
Algorithm during the COVID Pandemic in UAE

Paper 2

Sentiment Analysis Predictions in Digital Media Content using NLP Techniques

(Abdulrahman Radaideh1, 2023)

Table of Contents

Contents

Implementation of Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence	1
Abstract.....	3
List of Figures.....	16
List of Tables	18
Chapter I. Introduction.....	20
Diverse and Nuanced Human Expressions:	31
Rapid Evolution of Language:	32
Sheer Volume of Digital Data:	33
The Need for Automated Solutions:.....	33
Research Aim.....	44
Chapter II. Literature review	54
2.2.1.1 Key Methodologies and Technologies in Sentiment Analysis	56
Chapter III. Methodology	77
Tools and Resources	90
Machine learning-Based Models	93
Features Selection	94
1. Data pre-processing	94
2. Feature Extraction	96
Summary of Results	107
Chapter IV. Implementation.....	130
4.3.2.1 Introduction of New Data Sources	137
4.3.2.2 Modified Data Collection Techniques.....	138

4.3.3.1 Challenges Faced in Data Collection.....	138
4.3.3.2 Enhancements Implemented to Mitigate Challenges	139
Chapter V. Discussion	176
Chapter VI.....	189
Conclusion	189
References.....	194
APPENDICES	206

LIST OF ABBREVIATIONS

AI - Artificial Intelligence

API - Application Programming Interface

AUC - Area Under the Curve

BERT - Bidirectional Encoder Representations from Transformers

CNN - Convolutional Neural Network

CSV - Comma-Separated Values

DOI - Digital Object Identifier

DMC - Digital Media Content

FB - Facebook

GPT - Generative Pre-trained Transformer

HTML - HyperText Markup Language

IMDb - Internet Movie Database

ISBN - International Standard Book Number

JSON - JavaScript Object Notation

LDA - Latent Dirichlet Allocation

ML - Machine Learning

NLP - Natural Language Processing

PDF - Portable Document Format

QM - Quality Management

ROC - Receiver Operating Characteristic

RNN - Recurrent Neural Network

SP - Spotify

TW – Twitter

URL - Uniform Resource Locator

YT - YouTube

List of Figures

Figure 1 technique of Mining	31
Figure 2Multi head attention.....	32
Figure 3 framework.....	55
Figure 4 Content analysis in health care research.....	57
Figure 5 Data mining for diabetic data	60
Figure 6 Mobile manifestation (Msughter, 2023).....	64
Figure 7 bag of words	67
Figure 8 Digital twin.....	68
Figure 9 Opportunities framework for contributions to knowledge	75
Figure 10 conference paper detail.....	81
Figure 11 Research Procedure for Naive Bayes and RNN Algorithms for Real-Time Sentiment Analysis	82
Figure 12 Proposed Framework: A Novel Approach to Predict the Real-Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE	84
Figure 13 Measurement of positive and negative recall of the technique	86
Figure 14 Measurement of positive and negative precision of the technique.....	86
Figure 15 Accuracy of the methods RNN and NB	87
Figure 16 – NB Sentiment breakdown.....	87
Figure 17 General Framework Sentiment Analysis Predictions in Digital Media Content Using NLP Technique.....	90

Figure 18 Framework of Model	94
Figure 19 Data pre processing	95
Figure 20 Confusion matrix	100
Figure 21 Framework for proposed Model	102
Figure 22 . Plot of Results of RNN Model	105
Figure 23 Positive sentiments tweets	106
Figure 24. Negative sentiments tweets	106
Figure 25 our novel framework for Multimodal Sentiment Analysis:	120
Figure 26 Textual sentiment analysis techniques	121
Figure 27 framework to collect text from social medias	122
Figure 28video to audio then text	123
Figure 29 Chatgpt AI	131
Figure 30 General framework for case study.....	131
Figure 31 Novel method of case study	132
Figure 32 Most liked words on digital media about ChatGPT from January to March 2023.	133
Figure 33 Data preprocessing and analysis.....	134
Figure 34 after null values	143
Figure 35 Addressing Incorrectly Entered Values and cleaned dataset.....	144
Figure 36 null values removed.....	145
Figure 37 data after null value	146
Figure 38 text sentiment analysis.....	147
Figure 39 Model implementation.....	148
Figure 40 Sentiment distribution	150
Figure 41 trends in tweets counts.....	153

Figure 42 Average tweet Distribution based on Hours in a Day	155
Figure 43 Average tweet Distribution based on a days of the week.....	156
Figure 44word count	157
Figure 45 prominent words.....	158
Figure 46 subjective inference about possible topics on positive tweets.....	162
Figure 47 possible tweets topics	163
Figure 48 Negative tweets	163
Figure 49 Negative tweet topics	164
Figure 50 possible neutral tweets.....	166
Figure 51 neutral tweet topic	167
Figure 52 Roc for Logistic regression	170
Figure 53 ROC for Naive Bayes	171
Figure 54 Roc for Random forest	172

List of Tables

Table 1Research Map	45
Table 2 Gap Analysis from previous research.....	71
Table 3 Performance Measure of NB	85
Table 4 Performance Measure of RNN	85
Table 5 Accuracy of methods	85
Table 6 Python Libraries.....	91
Table 7 Validation Accuracy	99
Table 8 Recall, Precision, F1 Score of all algorithms.....	101
Table 10 shows tweets, posts, or comments from Twitter, IMDb, Amazon, Spotify, Facebook, and a website video platform.....	135
Table 11 New Data Sources.....	137

Table 12 Modified Data Collection Techniques.....	138
Table 13 Challenges Faced in Data Collection.....	138
Table 14 Enhancements Implemented to Mitigate Challenges.....	139
Table 15 libraries used in this case study.....	140
Table 16 performance of ML/DL algorithms	168

Chapter I.

Introduction

1.1 Introduction

In recent years, the domain of sentiment analysis and opinion mining has gained immense prominence in the realm of digital media content. This research explores the utilization of an innovative opinion mining approach, powered by artificial intelligence (AI), to analyze sentiments within this dynamic landscape (Waqas, 2022). Understanding the significance of this research requires a deeper examination of the following aspects:

1.1.1 The Emerging Field of Opinion Mining:

Opinion mining, often referred to as sentiment analysis, involves the automated extraction and evaluation of sentiments, attitudes, and opinions expressed within digital media content, including text, audio, and video (Pang, 2008). In a world where information is overwhelmingly digitized and available at our fingertips, the need to decipher the sentiments embedded within this vast content has become increasingly crucial.

1.1.2 Timeliness and Relevance in the Digital Age:

In the digital age, where communication occurs at an unprecedented scale and speed, the ability to decipher the sentiments of individuals, communities, and the general public is more pertinent than ever before. The real-time nature of digital media content means that understanding opinions and sentiments swiftly is essential for making informed decisions in various fields, such as business, politics, and public perception management.

1.1.3 Growing Influence of Online Content and User-Generated Opinions:

The rise of social media, online forums, blogs, and news websites has given individuals a powerful platform to express their opinions and emotions openly. These user-

generated opinions play a pivotal role in shaping public discourse, consumer choices, political narratives, and brand reputations (Albahri, 2020). Consequently, understanding the opinions and sentiments embedded in digital media content is no longer an option but a necessity.

In light of these factors, this research delves into the development and implementation of a novel opinion mining approach, driven by artificial intelligence, to effectively capture, analyze, and utilize sentiments expressed within the ever-expanding digital content landscape. This thesis endeavors to explore how AI technologies can enhance our understanding of online opinions, offering valuable insights for decision-makers in an era where digital media content exerts an unprecedented influence on our lives (Dogan, 2021).

1.2 Background

Opinion mining, also known as sentiment analysis, has evolved significantly over the years, with a rich history closely intertwined with the digital media content landscape (Martínez-Plumed, 2019). This historical perspective provides valuable context and underlines the importance of our current research (Alotaibi, 2023).

1.2.1 Historical Evolution of Opinion Mining:

Opinion mining finds its roots in the early efforts to understand and analyze text-based opinions and sentiments. It emerged as a response to the growing need to extract meaningful insights from the surging volumes of digital content available on the internet (Musleh, 2023). The field has witnessed remarkable progress since its inception.

1.2.2 Key Milestones and Influential Research Papers:

Several key milestones and influential research papers have significantly shaped the field of opinion mining. The seminal work of (Bock, 2019) in their paper "Opinion Mining and Sentiment Analysis" published in 2019 marked a turning point in the development of

sentiment analysis methodologies. Their comprehensive review of sentiment analysis techniques laid the foundation for subsequent research in the field.

Additionally, the advent of large-scale sentiment analysis datasets, such as the Sentiment140 dataset and the IMDB movie reviews dataset, has been instrumental in advancing sentiment analysis techniques (Alomari, 2023). These datasets provided researchers with the means to develop and benchmark new algorithms and models.

1.2.3 Development of AI Technologies Relevant to Opinion Mining:

The evolution of opinion mining is closely linked with advancements in artificial intelligence technologies. Natural Language Processing (NLP), a subfield of AI, has played a pivotal role in enabling the automated analysis of sentiments within text data. Early NLP techniques focused on rule-based approaches, but the field later embraced machine learning algorithms, including support vector machines and neural networks, which significantly improved sentiment analysis accuracy (Polat, 2020).

Recent years have witnessed the rise of deep learning models, such as recurrent neural networks (RNNs) and transformers, which have pushed the boundaries of sentiment analysis. These models excel in capturing the nuances of human language and have become a cornerstone in the development of state-of-the-art sentiment analysis systems.

In the contemporary digital era, the proliferation of online platforms and the ease of content creation have led to an exponential growth in digital media content. This content spans a wide spectrum, including news articles, blogs, social media posts, reviews, comments, and user-generated content on platforms like Twitter, Facebook, Instagram, and YouTube. In this vast sea of information, opinions, emotions, and sentiments are expressed on a diverse range of topics, from products and services to political events, social issues, and entertainment (Imran, 2023).

The ability to extract, analyze, and interpret these opinions, sentiments, and emotions expressed in digital media content is crucial for various stakeholders, including businesses, government agencies, researchers, and the general public. This need has given rise to the field of opinion mining, also known as sentiment analysis, which seeks to develop methods and techniques for understanding, categorizing, and summarizing the sentiments expressed in text, multimedia, or other forms of digital content (Azam, 2023). The insights derived from opinion mining can be invaluable for decision-making, brand management, product development, understanding public sentiment, and much more.

Traditional opinion mining approaches often relied on rule-based systems or lexical resources like sentiment lexicons. These approaches, while useful to some extent, struggled to cope with the nuances, context, and evolving language used in digital content. The need for more sophisticated techniques led to the emergence of machine learning and natural language processing (NLP) approaches, which marked a significant shift in the field of opinion mining.

Machine learning models, particularly supervised and unsupervised learning methods, have shown promise in automatically categorizing text as positive, negative, or neutral, as well as in identifying more nuanced sentiments and emotions. NLP techniques, including part-of-speech tagging, syntactic parsing, and named entity recognition, have been integrated into opinion mining pipelines to enhance the accuracy of sentiment analysis.

However, as the digital landscape continually evolves, opinion mining faces new challenges. The rise of multimedia content, the prevalence of slang, dialects, and the ever-changing nature of language make sentiment analysis a dynamic and evolving field.

Moreover, the enormous scale of digital media content, often referred to as "big data," necessitates scalable and efficient opinion mining approaches.

In response to these challenges, artificial intelligence (AI) has emerged as a potent ally in opinion mining. AI, particularly deep learning models like neural networks and recurrent neural networks (RNNs), has shown the ability to capture complex patterns in language, leading to significant advancements in sentiment analysis (Al-Shareeda, 2023).

With this background, the need to explore and implement a novel opinion mining approach using artificial intelligence becomes evident. This research aims to harness the capabilities of AI to address the shortcomings of traditional methods and tackle the ever-evolving landscape of digital media content. It seeks to develop innovative techniques for opinion mining that can handle multimodal data, provide real-time insights, and be customized for specific domains while also addressing ethical considerations (Butt, 2023). By doing so, it aims to contribute to the growing body of knowledge in this field and offer practical solutions to the challenges presented by the digital age.

The growth of digital media content has been fueled by the ease of content creation, the ubiquity of internet access, and the advent of mobile devices. These factors have transformed the way people engage with information and express their views on a multitude of subjects. Consequently, opinion mining has evolved from being a niche area of research to a critical component of modern information processing. Its applications are widespread, impacting various sectors, including but not limited to business, politics, healthcare, marketing, and social sciences.

In business, opinion mining plays a pivotal role in understanding customer sentiment, gauging the reception of products and services, and optimizing marketing strategies. For instance, businesses can utilize sentiment analysis to assess how consumers perceive their products, identify areas for improvement, and tailor marketing campaigns to specific demographics or regions. Customer feedback, whether it comes from online reviews, social media mentions, or customer support interactions, contains invaluable insights that can directly impact business decisions and profitability.

In the realm of politics and governance, opinion mining can help policymakers and government agencies assess public sentiment on various issues. Understanding the mood of the electorate, tracking public response to government policies, and identifying emerging concerns can influence decision-making and communication strategies. Moreover, opinion mining can aid in early detection of public crises and sentiment shifts, allowing for proactive crisis management and responsive governance.

The healthcare industry benefits from opinion mining by monitoring patient experiences and feedback. Through the analysis of patient reviews, comments on healthcare providers, and social media discussions, healthcare organizations can improve the quality of care and patient satisfaction. Rapid identification of potential issues or trends in patient feedback can lead to quicker improvements in healthcare services and patient outcomes.

The marketing industry has long been a pioneer in leveraging opinion mining. By analyzing social media conversations, product reviews, and other forms of customer feedback, marketers can gain insights into consumer preferences, market trends, and

competitive intelligence. This knowledge can be utilized to create more effective advertising campaigns, develop new products, and gain a competitive advantage.

The research community, including social scientists, linguists, and computer scientists, has also benefited significantly from the growth of opinion mining. It has opened up new avenues for studying public discourse, societal trends, and the impact of language on public opinion. Researchers can employ sentiment analysis to investigate public responses to political events, social movements, and cultural phenomena, shedding light on the dynamics of public discourse and the factors that influence it.

While opinion mining has shown its potential in various domains, the traditional approaches have struggled to adapt to the dynamic and multifaceted nature of digital media content. The earliest methods predominantly relied on rule-based systems, where predefined sets of linguistic rules and sentiment lexicons were used to classify text as positive, negative, or neutral. These approaches had limitations, particularly when faced with the complexities of natural language, context-dependent sentiment, and the ever-evolving language used in digital content.

The advent of machine learning, and more recently deep learning, has significantly improved sentiment analysis. Machine learning models, both supervised and unsupervised, have demonstrated the ability to learn complex patterns in language and provide more nuanced sentiment classification. For instance, supervised learning models are trained on labeled datasets, enabling them to classify text into sentiment categories. Unsupervised learning models, on the other hand, identify patterns and clusters within the data without

requiring labeled examples. Both approaches have been instrumental in advancing sentiment analysis by making it more adaptive and less reliant on manual rule-based systems.

Natural language processing (NLP) techniques have been integrated into opinion mining workflows to enhance the accuracy of sentiment analysis. Part-of-speech tagging, syntactic parsing, and named entity recognition are some of the NLP techniques employed to better understand the linguistic structure and context of text. This integration has enabled sentiment analysis models to consider the grammatical structure of sentences and the roles of entities within them, providing a more sophisticated analysis of sentiment.

However, even with these advancements, several challenges remain. The landscape of digital media content continues to evolve. The rise of multimedia content, such as images, videos, and audio, has added new dimensions to sentiment analysis. Analyzing multimodal content requires techniques that can combine information from different sources, raising questions about feature extraction, integration, and interpretation. Additionally, the analysis of non-textual data, such as images and videos, introduces complexities related to image recognition, object detection, and scene understanding.

The dynamic nature of language, with the emergence of slang, jargon, and dialects, further complicates sentiment analysis. Keeping up with the ever-changing vocabulary and linguistic nuances in various domains poses a substantial challenge. Sentiments expressed through sarcasm, irony, or humor can be particularly difficult to detect, as they often rely on contextual cues that may not be captured by traditional sentiment analysis methods.

Furthermore, the sheer scale of digital media content presents another significant obstacle. The term "big data" aptly describes the volume of information generated and shared on the internet daily. Efficiently processing and analyzing this vast amount of data requires scalable solutions that can handle high throughput and large datasets.

The need for real-time sentiment analysis is another dimension that arises from the dynamic nature of digital media. Opinions and trends can change rapidly, and businesses and organizations require the ability to monitor and respond to shifts in sentiment in real-time. Traditional batch processing methods may not be adequate for this purpose, leading to the development of real-time opinion tracking systems.

Customization for specific domains is yet another challenge. Different domains, such as politics, product reviews, financial markets, and social issues, have unique vocabularies, contexts, and sentiment dynamics. A one-size-fits-all sentiment analysis model may not be effective in capturing the nuances of domain-specific language and sentiment. Adapting sentiment analysis techniques to these specific domains is necessary to ensure that the results are relevant and actionable.

As opinion mining continues to evolve, ethical considerations have also gained prominence. The responsible use of AI and machine learning in opinion mining must address issues related to privacy, bias, fairness, and transparency. These concerns become increasingly important as AI-powered sentiment analysis systems are integrated into real-world applications, where they can influence decisions and public opinion.

Given the challenges and opportunities outlined in this background, the role of artificial intelligence (AI) emerges as a promising approach to enhance opinion mining. AI, particularly deep learning models, has demonstrated a remarkable ability to capture intricate language patterns and adapt to evolving linguistic trends. These models have significantly improved the accuracy and depth of sentiment analysis.

Deep learning models, such as neural networks and recurrent neural networks (RNNs), have the capacity to process sequential data, making them well-suited for analyzing text. They can capture long-range dependencies in language and identify context-dependent sentiments. For instance, RNNs, with their ability to retain information over long sequences, can be employed to analyze text in a way that accounts for the influence of words further back in the text on the sentiment expressed in a current sentence or phrase.

With the advent of transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), AI has shown even more potential for capturing contextual information in text. These models employ attention mechanisms to weigh the importance of different words in a sentence, allowing them to understand the nuanced relationships between words and phrases. As a result, they excel in various NLP tasks, including sentiment analysis.

Artificial intelligence also offers the ability to process non-textual data, such as images and videos. Image recognition, object detection, and scene understanding can be integrated with AI-based sentiment analysis to extract sentiment from visual content. For example, in a product review, AI can analyze both the textual review and accompanying images to provide a holistic assessment of the user's sentiment.

To address the challenges of real-time analysis, AI-driven opinion mining can utilize online learning and streaming data processing techniques. These approaches allow for the continuous monitoring of digital media content and the immediate detection of sentiment shifts. Real-time opinion tracking systems can keep organizations informed about emerging trends and issues in real-time, enabling rapid response and informed decision-making.

Customization for specific domains becomes more achievable with AI. Machine learning models can be fine-tuned on domain-specific data to adapt to the particular language and sentiment patterns in that domain. The flexibility of AI models and the availability of pre-trained models for various languages and domains make domain-specific sentiment analysis a realistic and practical goal (Sarker, 2023)

Ethical considerations are also an integral part of the AI-driven opinion mining approach. AI models can be designed and trained with ethical guidelines in mind to mitigate privacy concerns, address biases, and ensure transparency in decision-making. Responsible AI practices should be adhered to throughout the development and deployment of AI-based opinion mining systems.

1.3 Overview of the Problem of Opinion Mining on Digital Media Content:

In the digital age, understanding and harnessing the sentiments and opinions embedded in the vast landscape of digital media content is a multifaceted challenge that our research endeavors to address.

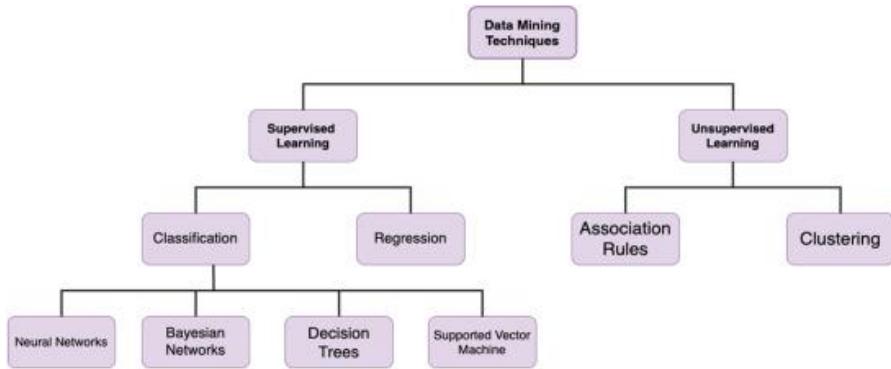


Figure 1 technique of Mining

This section offers a detailed understanding of the problem at hand and why it poses a unique set of challenges.

1.3.1 The Challenge of Opinion Mining:

Opinion mining, often referred to as sentiment analysis, is a multifaceted process that involves the automatic identification and categorization of opinions, sentiments, and emotions expressed in digital media content. While the overarching concept may appear straightforward, delving deeper reveals the intricacies and challenges that accompany this endeavor. In this explanation, we will explore the complexities of opinion mining, the nuances of human expressions, the ever-evolving nature of language, and the sheer volume of data generated in digital spaces.

Diverse and Nuanced Human Expressions:

One of the foremost challenges in opinion mining lies in the diversity and nuance of human expressions. Humans communicate their thoughts and emotions through a wide range of words, phrases, tones, and body language. These expressions can be straightforward or incredibly subtle, requiring a deep understanding of context and cultural factors to decipher accurately.

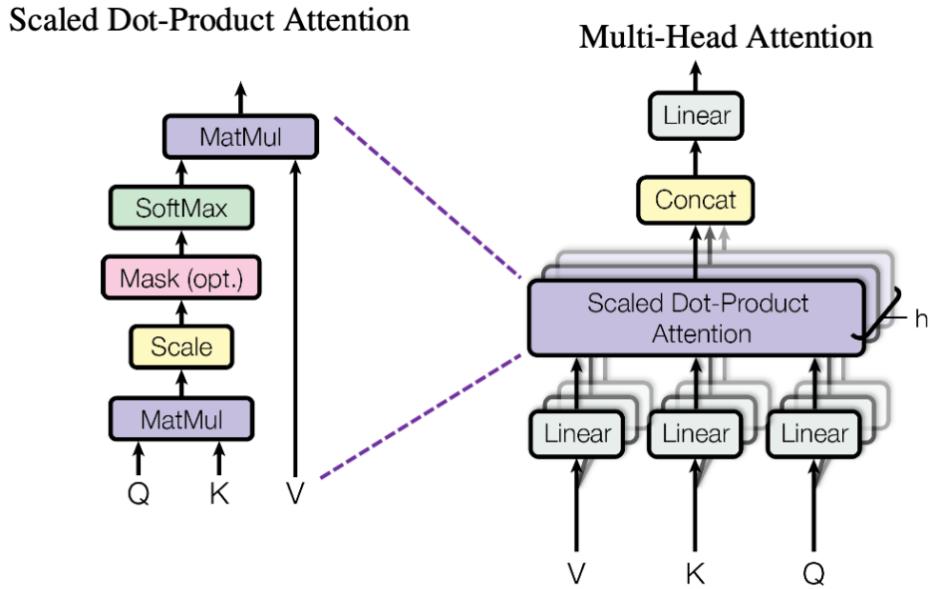


Figure 2Multi head attention

Consider the statement, "The movie was good." On the surface, it appears positive. However, the interpretation can change drastically depending on the context. If this statement is found in a movie review blog post, it's likely a positive sentiment. But if it's on a blog discussing nutrition, the sentiment may be more neutral, focusing on the nutritional value of popcorn at the cinema. Context, therefore, plays a pivotal role in understanding the sentiment behind a statement.

Additionally, human expressions are highly influenced by cultural and demographic factors. What is considered a positive sentiment in one culture may not be perceived the same way in another. Slang, idioms, and colloquialisms further complicate the interpretation of opinions and sentiments.

Rapid Evolution of Language:

Language is not static but continuously evolves with societal and technological changes. This dynamic nature of language presents a challenge in opinion mining. New words, phrases, and expressions constantly emerge, and their meanings can shift over time.

For example, the word "cool" has evolved from its original temperature-related meaning to become synonymous with "impressive" or "stylish." In digital spaces, language often evolves rapidly, with new words and phrases emerging due to pop culture, technological advancements, or social trends. Keeping pace with these linguistic shifts is essential for accurate sentiment analysis.

Moreover, the rise of internet culture, memes, and social media has introduced entirely new forms of language and communication, often characterized by humor, irony, and hyperbole. Understanding and interpreting these expressions require a deep understanding of internet culture and context.

Sheer Volume of Digital Data:

The digital age has given rise to an unprecedented volume of content in various forms: text, audio, images, and video. Social media platforms, blogs, news websites, forums, and online communities generate an overwhelming amount of data every second. This deluge of information makes the task of opinion mining challenging.

To put this into perspective, consider that every minute, thousands of tweets are posted on Twitter, millions of Facebook posts are shared, and an enormous amount of content is uploaded to YouTube. Processing and analyzing this sheer volume of data in real-time or near real-time is a formidable undertaking.

Moreover, not all digital content is text-based. Videos and images often convey sentiments and opinions through visual cues, facial expressions, and audio elements. Opinion mining algorithms must be versatile enough to handle multiple content formats.

The Need for Automated Solutions:

Given the complexities of opinion mining, manual analysis of such vast and varied data is practically impossible. This is where the role of artificial intelligence, machine

learning, and natural language processing becomes critical. These technologies empower automated sentiment analysis at scale.

Machine learning algorithms can be trained to recognize patterns and context, enabling them to accurately categorize sentiments in text, audio, and visual content. Natural language processing techniques allow machines to understand and interpret human language, even in its diverse and evolving forms.

1.4 The Role of Artificial Intelligence:

Artificial intelligence, particularly natural language processing (NLP) and machine learning, has been instrumental in advancing the field of opinion mining. NLP techniques enable machines to understand and analyze human language. They can identify sentiment-bearing words, phrases, and expressions in text, allowing for sentiment classification.

Machine learning models, such as support vector machines, decision trees, and neural networks, are used to train algorithms to recognize sentiment patterns in data. These models can generalize from labeled training data to make predictions on new, unseen content. As more data is processed, machine learning models can adapt and improve their accuracy.

In recent years, deep learning models, particularly recurrent neural networks (RNNs) and transformer-based architectures, have shown remarkable success in capturing the nuances of human language. These models excel at understanding context, sarcasm, idiomatic expressions, and other subtle elements of communication.

The Importance of Context:

Context is a vital component of sentiment analysis. It involves understanding the circumstances, the subject matter, and the cultural and demographic factors surrounding a piece of content. Without context, sentiment analysis can produce inaccurate results.

For instance, the statement "This phone is a bomb!" may be considered positive in a casual conversation among friends but could be negative when discussing product reviews due to the negative connotation of the term "bomb." Recognizing these subtleties is a key challenge in sentiment analysis.

Challenges Beyond Text:

Opinions and sentiments are not limited to text. Visual and auditory content, such as images, videos, and audio recordings, also convey emotions and opinions. Analyzing multimedia content poses additional challenges, as sentiment may be expressed through facial expressions, tone of voice, or visual elements. Deep learning models, capable of image and audio analysis, are employed to tackle these challenges.

In conclusion, opinion mining, or sentiment analysis, presents a complex and multifaceted challenge due to the diverse and nuanced nature of human expressions, the rapid evolution of language, and the sheer volume of data generated in digital spaces. However, the advancement of artificial intelligence, especially in the domains of natural language processing and machine learning, has paved the way for automated solutions that can make sense of this complex landscape. Despite the challenges, opinion mining plays a crucial role in understanding public sentiment, consumer preferences, and trends in various domains, from business to politics, making it a field of significant relevance in the digital age

In the digital age, understanding and harnessing the sentiments and opinions embedded in the vast landscape of digital media content is a multifaceted challenge that our research endeavors to address. This section offers a detailed understanding of the problem at hand and why it poses a unique set of challenges.

The Complexity of Digital Media Content:

Digital media content encompasses a vast array of formats, including text, audio, and video. These content types are generated on an extensive array of platforms, such as social media networks, blogs, news websites, and multimedia sharing platforms. This diversity in content and platforms presents both an opportunity and a challenge.

Diverse Formats of Digital Media Content:

Digital media content spans a wide spectrum of formats, each offering a unique mode of communication. These formats include text, audio, and video, with each presenting distinct challenges and opportunities for analysis.

- **Text-Based Content:** Textual content, such as articles, blog posts, social media updates, and comments, is a fundamental component of digital media. This format is highly adaptable, allowing for the expression of opinions, ideas, and information in a structured manner. However, text can be highly nuanced, and interpreting sentiments and emotions embedded within it requires an understanding of context, tone, and idiomatic expressions. The sheer volume of textual data generated daily across the internet presents a significant challenge for opinion mining, as it necessitates efficient methods for processing and analysis.
- **Audio Content:** With the proliferation of podcasts, audiobooks, music streaming, and voice assistants, audio content has gained prominence in the digital space. Sentiments and emotions can be conveyed through vocal tone, inflection, and speech patterns. Analyzing audio content poses unique challenges, as it requires the conversion of spoken language into text for sentiment analysis. Automatic speech recognition (ASR) and natural language processing (NLP) technologies play a crucial role in this process.
- **Video Content:** Videos, including vlogs, YouTube channels, television shows, movies, and user-generated content, provide a rich medium for expressing opinions

and sentiments. In addition to spoken language, video content incorporates visual and auditory elements, making sentiment analysis more complex. Understanding the emotions conveyed through facial expressions, body language, and visual storytelling adds layers of complexity to opinion mining. Moreover, extracting sentiments from video content requires transcribing spoken words and analyzing the accompanying visual and auditory cues, necessitating advanced multimedia sentiment analysis techniques.

A Plethora of Platforms:

Digital media content is not confined to a single, monolithic platform but is disseminated across an extensive array of online spaces. These platforms serve as vehicles for the creation, distribution, and consumption of content. Some of the notable platforms encompass:

- **Social Media Networks:** Platforms like Facebook, Twitter, Instagram, and TikTok host a vast ecosystem of content, including text, images, and videos. Users share their thoughts, experiences, and opinions, making social media a fertile ground for sentiment analysis. The challenge here is not only the volume of content but also the speed at which it is generated. Sentiments can change rapidly in response to real-time events, trends, and conversations.
- **Blogs:** Blogging platforms, such as WordPress and Medium, host textual content created by individuals, organizations, and experts in various domains. Bloggers share their opinions, reviews, and experiences, making this format a valuable source for opinion mining. However, the diversity of blogging topics and writing styles necessitates robust sentiment analysis techniques to cater to different contexts and subject matter.

- **News Websites:** News outlets and online publications publish a continuous stream of news articles and reports. Analyzing the sentiments expressed in news articles can provide insights into public perception and media bias. News content is often characterized by a formal and objective tone, making sentiment analysis challenging due to its distinct style.
- **Multimedia Sharing Platforms:** Platforms like YouTube, Vimeo, and SoundCloud host a plethora of audio and video content. Users upload videos, music, podcasts, and more, expressing opinions and emotions through both spoken words and visual storytelling. The challenges in sentiment analysis here extend beyond the diversity of content to the need for effective transcription, audio analysis, and image recognition.

Opportunities and Challenges:

The diversity of content formats and platforms within the digital media landscape presents both opportunities and challenges for opinion mining.

- **Opportunities:**
 - **Rich Data Sources:** The availability of diverse content formats provides a wealth of data for sentiment analysis. This data can be harnessed to gain insights into public sentiment, consumer preferences, and emerging trends.
 - **Comprehensive Understanding:** Analyzing a variety of content types enables a comprehensive understanding of opinions and emotions. Combining text, audio, and video analysis allows for a more holistic view of sentiment within the digital sphere.
 - **Real-Time Insights:** Many digital media platforms operate in real time, making it possible to capture and analyze sentiments as they evolve. This real-time analysis is invaluable for tracking public sentiment during events, product launches, or crises.

- **Challenges:**
 - **Data Volume:** The sheer volume of digital content generated daily poses a logistical challenge. Efficient methods for data collection, storage, and analysis are essential to handle this data deluge.
 - **Multimodal Analysis:** Analyzing audio and video content requires the integration of text analysis with audio and visual processing. Developing and deploying multimodal sentiment analysis systems is a complex undertaking.
 - **Cultural and Contextual Variation:** Sentiments are expressed differently in various cultures and contexts. Understanding the cultural and contextual nuances of sentiments is crucial for accurate analysis.
 - **Evolving Platforms:** Digital platforms are continually evolving. New platforms and communication trends introduce new challenges for sentiment analysis. Staying up to date with emerging platforms and language trends is imperative.

Digital media content is a complex and dynamic ecosystem that encompasses a variety of formats and platforms. Sentiments and opinions are expressed through text, audio, and video, each with its unique characteristics. While this diversity provides opportunities for comprehensive sentiment analysis, it also poses challenges in terms of data volume, multimodal analysis, cultural context, and the ever-changing digital landscape. To harness the potential of this digital content ecosystem, sentiment analysis techniques need to be adaptable, efficient, and capable of providing insights in real time.

The Opportunity:

In the digital age, the vast and diverse landscape of digital media content serves as a veritable treasure trove of opinions, offering a unique and unprecedented opportunity to gauge public sentiment, collect valuable consumer feedback, discern political leanings, and

track evolving social trends. The potential benefits span a wide spectrum, from aiding businesses in gaining a profound understanding of customer preferences and improving their products and services to assisting governments in assessing public opinion on policies and complex societal issues. It's a dynamic and multifaceted arena where the analysis of opinions can yield insights that shape decisions, fuel innovations, and foster a deeper connection with individuals, communities, and the world at large. This opportunity transcends industries, and it's crucial to delve into the myriad ways in which opinion mining can contribute to the betterment of society, commerce, and governance.

The Challenge:

However, this vast landscape is characterized by its unstructured and ever-evolving nature. The challenges lie in processing, categorizing, and extracting meaningful insights from this abundance of data. Opinions and sentiments are often expressed in diverse and subtle ways, making their interpretation a non-trivial task.

Moreover, the dynamic nature of language and the constant emergence of new terminologies and slang present an ongoing challenge. Additionally, the rapid proliferation of multimedia content, including images, audio, and video, adds an extra layer of complexity to opinion mining, as sentiments are conveyed not only through text but also through visual and auditory cues.

Our research seeks to confront these challenges by harnessing the power of artificial intelligence to make sense of this unstructured and ever-evolving digital media content. By developing and implementing a novel opinion mining approach, we aim to advance the field, providing more accurate and efficient methods for extracting valuable insights from this rich but complex data landscape.

1.Role of Digital Media in AI

Digital media content, spanning text, images, videos, and audio, plays a pivotal role in the information landscape of the modern world. The explosive growth of digital content on the internet, coupled with the advent of Artificial Intelligence (AI), has given rise to new possibilities and challenges. AI has become a transformative force in the creation, curation, and analysis of digital media content. This article explores the multifaceted role of AI in digital media content, examining its impact on content creation, content recommendation, content moderation, and content analysis.

1. Content Creation and Generation

AI is increasingly being employed in the generation of digital media content. Natural Language Processing (NLP) models, like GPT-3 and BERT, can produce human-like text, allowing for automated content generation. This has applications in areas such as journalism, where AI can assist in drafting news reports and generating articles at a remarkable speed.

In addition to text, AI has also made significant strides in generating visual and audio content. Generative Adversarial Networks (GANs) can create realistic images and videos. Deep learning models are used to synthesize human-like voices and music compositions. These AI-generated content forms are not only impressive but also practical for various creative and data-driven tasks.

2. Content Recommendation and Personalization

AI algorithms underlie content recommendation systems that have become integral to platforms like Netflix, Amazon, and social media networks. These recommendation engines

analyze user behavior, preferences, and past interactions to suggest content that users are likely to find engaging. By doing so, they enhance user experience, increase engagement, and contribute to content discovery.

Content personalization goes beyond just suggesting movies or products; it extends to tailoring news, articles, and information to individual users' interests. This can be seen in news aggregators that use AI to curate news articles based on the user's past reading habits and preferences. As a result, users are presented with content that aligns with their beliefs and interests, potentially creating information bubbles but also ensuring relevance and engagement.

3. Content Moderation and Safety

Digital media platforms are grappling with the challenge of moderating content at scale, ensuring that it adheres to community standards and legal regulations. AI-driven content moderation systems employ computer vision and NLP to scan images, videos, and text for harmful or inappropriate content. Such systems are vital for preventing the spread of hate speech, misinformation, and graphic content. AI plays a critical role in automating the content moderation process, assisting human moderators, and maintaining a safer online environment.

4. Content Analysis and Insights

Digital media content offers a wealth of information, and AI is instrumental in extracting insights from this content. Sentiment analysis, a subfield of NLP, is employed to

gauge public sentiment and opinions expressed in text. This has applications in market research, brand management, and public opinion tracking.

Content analysis can also involve categorizing and tagging content, as seen in e-commerce sites where AI assigns product categories and tags based on image recognition. In academic research, AI-driven content analysis helps researchers to analyze large corpora of text efficiently.

5. Content Translation and Multilingual Capabilities

The global nature of digital media content requires translation and localization to reach diverse audiences. AI-driven machine translation models, such as Google Translate, employ neural networks to provide fast and reasonably accurate translations across multiple languages. These models enable content creators to reach a broader global audience without the need for extensive manual translation efforts.

6. Content Enhancement and Editing

AI tools are increasingly used to enhance digital media content. Image and video editing software often incorporate AI features to improve image quality, remove imperfections, or add creative effects. AI-powered grammar and style checkers are valuable for writers, helping to improve the clarity and coherence of written content.

7. Content Accessibility

AI has made significant contributions to improving content accessibility. For example, AI-driven automatic speech recognition (ASR) technology is used to provide real-time captions for live events and videos, making content more accessible to individuals with hearing impairments. Similarly, AI-based text-to-speech (TTS) systems convert text into spoken words, aiding those with visual impairments.

8. Content Verification and Fact-Checking

The rapid spread of misinformation and fake news in the digital age has necessitated the development of AI tools for content verification and fact-checking. AI algorithms can analyze the credibility and authenticity of news articles and social media content, helping users make more informed decisions about the information they encounter.

9. Content Distribution and Optimization

AI is integral to content distribution strategies, helping content creators reach their target audience. AI-powered algorithms identify the most opportune times to post content on social media for maximum visibility and engagement. Moreover, AI-driven SEO tools analyze search trends and help optimize content for search engines, ensuring that it ranks higher in search results.

Research Aim

The research aims to perform novel sentimental analysis of users reviews on twitter digital media content in various forms (Video, Audio, Image, Text).

1.4 Research Objectives

To achieve the set research aim, the following objectives identified:

Objective 1: To justify the Importance of implementing sentiment analysis in social media platforms.

Objective 2: To build knowledge and investigate the recent literature about the research topic.

Objective 3: To propose a new approach to perform the prediction of the sentiments from social media users reviews by:

- Finding sentiments in text.
- Converting video and audio data into text
- Clustering of positive, negative and neutral tweets.

1.5 Research Questions

1. Why it is important to do the sentimental analysis? What is the existing methods to do the sentiment analysis and how we can do better than them?
2. Where to collect the data?
3. How to collect the data?
4. How to convert these user reviews from video, audio and image forms to text?
5. How to find the sentimental analysis on the text?
6. How to do in-depth label/classification of positive, negative and neutral tweets after the classification of sentimental analysis?

1.6 Research Map

Table 1Research Map

Research Problem	Aim	Objectives	Questions	Background Theory / Methodology
The other researches models have limitation only to text, the proposed research to include the image, video, facial expression, emoji and frames.	To perform a novel sentimental analysis in twitter digital media content users reviews in various forms (Video, Audio, Image, Text) to enable effective decision-making.	<p>Objective 1: To investigate/justify the Importance of implementing sentiment analysis in social media platforms.</p> <p>Objective 2: To build knowledge and investigate the recent literature about the research topic.</p> <p>Objective 3: To propose a new approach to perform the prediction of the sentiment analysis of the social media users reviews by:</p> <ol style="list-style-type: none"> 1. Collecting twitter user's reviews in various forms; Video, Audio, Image and Text in a specific location date and time. 2. Converting tweets from video, audio and image to text. 3. Finding sentiments on text. 4. Clustering of positive, negative and neutral tweets. 	Q1 Q1 Q2, Q3 Q4, Q5 Q6	Problem Formulation/ Investigate the recent literature Data Evaluation /Analysis and Interpretation Collecting data by using API and tools, "Tweepy" Mp4 Convertor OCR Convertor Video – Audio Converter ML techniques RNN,NB, LR ML techniques ELMO and CNN

A few years ago, this was difficult to identify and measure since the reviews have to be obtained from the public physically. A sample target population had to be identified and then interviewed for obtaining their feedback and reviews, however with the rise of AI algorithms, it is possible to receive the feedback from the public since the feedback is quite instantaneous and maybe also captured in real-time. It is also an open source information since it is available

to the public through social media platforms APIs. Social media has emerged as a vital platform of representing people's sentiment, boosting the requirements of data mining in the field of the sentiment analysis.

1.7 Research Contribution

The evolving landscape of sentiment analysis, as examined in the study titled "Implementation of Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence to aid in decision making and quality assurance" with a case study focusing on Chat GPT Sentiment Analysis, marks a crucial juncture for decision-making frameworks across sectors. Recognizing the shift in user-generated content, from conventional text-based tweets to a myriad of multimedia formats encompassing images, videos, and feedback, underscores the need for advanced sentiment analysis methodologies.

This research acknowledges the transformative potential of sentiment analysis, elucidating its instrumental role in refining product specifications, operational efficiency, and unlocking insights crucial for economic considerations. The study harnesses a machine learning approach to navigate sentiment analysis, thereby empowering organizations with enriched insights crucial for strategic decision-making processes.

A fundamental contribution of this research is the innovative approach towards extracting sentiments not only from conventional textual data but also from multimedia sources such as videos and audios. In an era where user reviews predominantly materialize in video formats on platforms like YouTube, the proposed methodology of converting multimedia reviews into textual data stands as a pioneering leap in sentiment analysis methodologies. This comprehensive approach elevates the depth and breadth of insights derived from digital media content.

The study incorporates data collection from an array of digital media platforms, including YouTube, IMDb, Spotify, Twitter, and Facebook, enriching the dataset with

diverse user-generated content. This amalgamation of data sources contributes to a holistic understanding of user sentiments across various platforms, paving the way for nuanced analyses and informed decision-making.

This research endeavor aspires to serve as a pivotal reference for organizational decision-makers, offering a foundational framework for evolving guidelines and methodologies in effective sentiment analysis. Moreover, the innovative techniques employed to handle diverse data formats underscore the study's commitment to addressing the evolving nature of user-generated content and sentiment analysis in the dynamic digital landscape.

1.8 Statement of the problem

In the era of pervasive digital media and online communication, the process of opinion mining and sentiment analysis has emerged as a critical area of study and application. Opinion mining, also known as sentiment analysis, involves the automated identification and classification of opinions, sentiments, and emotions expressed within digital media content. Understanding and effectively utilizing this process is of paramount importance due to the profound impact digital media content exerts on various facets of modern society.

1.8.1 The Pervasive Influence of Digital Media Content:

Digital media content, ranging from text, audio, to video, has become an integral part of our daily lives. The influence of digital media is not limited to one or a few sectors but transcends various domains, including business, politics, social dynamics, and communication. The pervasiveness of digital media content, especially through social media platforms, blogs, news websites, and multimedia sharing platforms, has given individuals a powerful and instantaneous medium to express their opinions and sentiments. This

transformation has in turn significantly altered the dynamics of communication, information dissemination, and public discourse.

1.8.2 The Challenges in Opinion Mining:

Despite the potential insights and opportunities that digital media content offers, opinion mining and sentiment analysis present formidable challenges. The complexity of human expressions, the rapid evolution of language, the sheer volume of data generated in digital spaces, and the diversity of content formats and platforms have made sentiment analysis a multifaceted task. Understanding the diverse and nuanced ways in which individuals express their thoughts and emotions in the digital realm is a challenge that necessitates a profound understanding of context, culture, and linguistic subtleties.

The rapid evolution of language in the digital age, characterized by the emergence of new words, phrases, and expressions influenced by pop culture, technology, and social trends, poses another layer of complexity. It requires sentiment analysis tools to be adaptable and capable of identifying and interpreting these emerging linguistic phenomena.

1.8.3 The Enormous Volume of Digital Data:

The digital age has ushered in an era of information abundance. The sheer volume of digital data generated in the form of text, audio, images, and video across a wide spectrum of platforms is staggering. Social media platforms, blogs, news websites, forums, and online communities generate a continuous deluge of data every second. The magnitude of this data presents logistical challenges in terms of collection, storage, and analysis. Managing and processing this data in real-time or near real-time is an imperative task in sentiment analysis.

Furthermore, the challenges of opinion mining extend beyond text-based content. Videos and images frequently communicate sentiments and opinions through visual cues, facial expressions, and audio elements. Sentiment analysis algorithms must be versatile enough to accommodate multiple content formats and modalities.

1.8.4 The Role of Artificial Intelligence in Opinion Mining:

To address these challenges, the field of opinion mining has turned to artificial intelligence (AI), particularly natural language processing (NLP) and machine learning. AI technologies enable the automation and scalability of sentiment analysis, making it possible to process vast amounts of data and adapt to evolving linguistic trends.

Natural language processing techniques empower machines to comprehend and interpret human language, accounting for its nuances and ever-evolving nature. Machine learning models, ranging from traditional support vector machines to state-of-the-art deep learning architectures, are employed to recognize sentiment patterns in data, whether textual, auditory, or visual.

1.8.5 The Significance of Context:

Context plays a pivotal role in sentiment analysis. It involves understanding the circumstances, the subject matter, and the cultural and demographic factors surrounding a piece of content. Sentiments are expressed differently in various cultures and contexts, making the interpretation of opinions a complex and context-dependent task.

For instance, the sentiment conveyed in the statement, "This is sick!" can vary greatly depending on whether it's part of a movie review, a discussion about health, or a casual conversation among friends. Recognizing these subtleties and contextual variations is crucial for accurate sentiment analysis.

1.8.6 The Multimodal Challenge:

Opinions and sentiments are not confined to text. Visual and auditory content, such as images, videos, and audio recordings, also communicate emotions and opinions. Analyzing multimedia content is challenging, as sentiments may be expressed through visual cues, facial expressions, tone of voice, or auditory elements. Effective sentiment analysis of multimedia

content requires the integration of text analysis with audio and visual processing, demanding advanced multimodal sentiment analysis techniques.

1.8.7 The Opportunities and Challenges in Opinion Mining:

The opportunities afforded by opinion mining are vast and wide-reaching. The digital landscape serves as a treasure trove of opinions, offering a unique opportunity to gauge public sentiment, collect consumer feedback, discern political leanings, and track evolving social trends. The potential benefits span a wide spectrum, from aiding businesses in understanding customer preferences and refining their products and services to helping governments assess public opinion on policies and complex societal issues. Sentiment analysis enables real-time insights into rapidly changing public sentiment, making it invaluable for tracking trends during events, product launches, or crises.

However, with these opportunities come a host of challenges. The sheer volume of data generated daily in digital media content necessitates efficient methods for collection, storage, and analysis. Multimodal sentiment analysis demands the integration of text, audio, and visual processing, requiring advanced technological capabilities. Cultural and contextual variations in sentiments introduce complexities that sentiment analysis tools must account for to ensure accuracy.

1.9 Thesis structure

Chapter 1: Introduction

This chapter provides an overview of the research, introducing the topic of opinion mining on digital media content using artificial intelligence. It discusses the significance of the research, outlines the challenges, and sets the stage for the subsequent chapters.

Chapter 2: Literature Review

In this chapter, we conduct a comprehensive review of the existing literature on opinion mining, sentiment analysis, and artificial intelligence in the context of digital media content. We explore key concepts, theories, methodologies, and notable research contributions in the field.

Chapter 3: Research Methodology

This chapter outlines the research methodology employed in the study. It discusses the data collection process, the design of experiments or case studies, the selection of machine learning or deep learning models, and the evaluation metrics used to measure the effectiveness of the opinion mining approach.

Chapter 4: Implementation

Here, we delve into the details of data collection from various digital media sources, including social media, news websites, and multimedia platforms. We describe the preprocessing steps for text, audio, and video content, such as text cleaning, audio transcription, and image processing.

Chapter 5: Discussion

This chapter introduces the novel opinion mining approach based on artificial intelligence. It elaborates on the model architecture, feature selection, and the techniques employed for sentiment analysis in text, audio, and video content. We discuss the algorithms and tools used for opinion classification. Here, we delve into the implications of our research findings. We discuss the practical applications of our opinion mining approach in different domains, including business, politics, and public perception. We also consider the ethical and societal implications of sentiment analysis in the digital age.

Chapter 6: Conclusion and Future Work

This final chapter summarizes the key findings, contributions, and limitations of our research. We propose areas for future research and offer concluding remarks on the significance of our opinion mining approach in the context of digital media content.

References

The reference section provides a comprehensive list of the sources cited throughout the thesis, adhering to the specific citation style used in the research.

Appendices

Any supplementary materials, code, data samples, or additional information are included in the appendices for reference and transparency.

This thesis structure provides a clear roadmap for the reader, outlining the flow of the research and the progression from introduction to conclusion. However, remember that the structure can be adjusted based on your specific research focus and supervisor's recommendations

Chapter II.

Literature review

2.1 Introduction

The literature review in this chapter serves as a critical exploration of existing knowledge, providing a comprehensive backdrop for the implementation of a novel opinion mining approach on digital media content using artificial intelligence (AI) in the realm of Quality Management Decision Making. This section aims to delineate the contextual foundation, identify key areas of research, and synthesize relevant literature that underpins the integration of AI-driven opinion mining in enhancing quality management strategies within various industries, particularly focusing on the dynamic landscape of digital media content analysis.

2.2 Understanding the Role of Opinion Mining/Sentiment Analysis and Artificial Intelligence in Quality Management

Opinion mining, or sentiment analysis, and artificial intelligence (AI) are integral components of quality management, playing key roles in understanding customer sentiments and ensuring product or service excellence (Cambria, 2013). Opinion mining involves extracting subjective information from textual data, enabling organizations to analyze customer feedback and identify quality issues. AI contributes by employing predictive analytics, process automation, and root cause analysis to enhance overall quality management. The integration of opinion mining and AI enables real-time monitoring, customer-centric quality improvement, and data-driven decision-making, fostering a proactive approach to identifying and addressing quality issues, ultimately leading to continuous improvement in products and services.

2.2.1 Defining Opinion Mining and Sentiment Analysis

The roots of sentiment analysis can be traced back to the early days of NLP and ML (Catelli, 2023). Early sentiment analysis efforts primarily focused on binary sentiment classification, distinguishing between positive and negative sentiments in textual data. As digital media evolved, sentiment analysis adapted to accommodate the nuanced and multifaceted nature of sentiments expressed in online content (Braig, 2023).

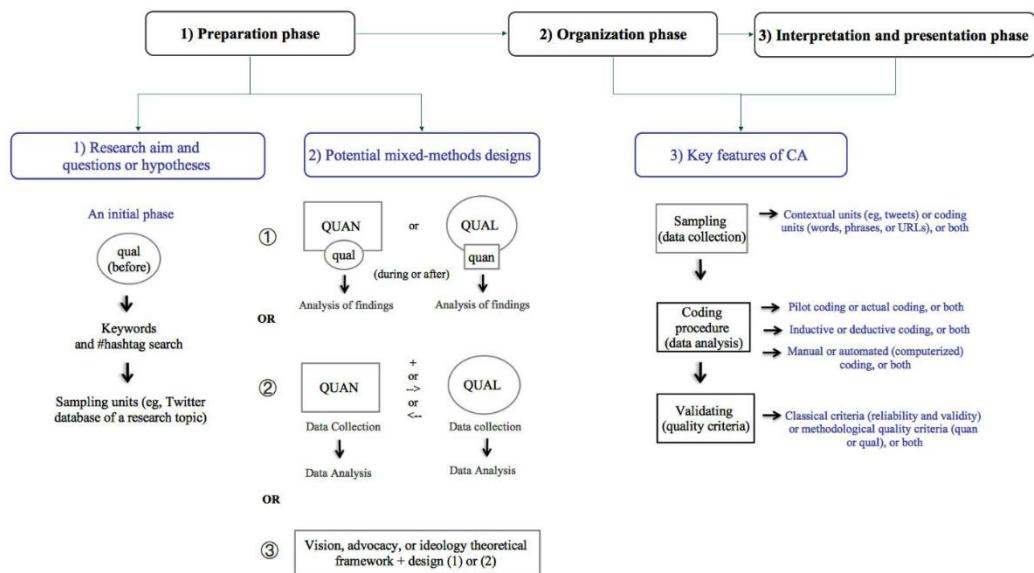


Figure 3 framework

The advent of social media platforms in the early 2000s marked a significant turning point (Carvache-Franco, 2023). Researchers and organizations recognized the potential of sentiment analysis to extract valuable insights from the vast volumes of user-generated content on platforms like Twitter and Facebook. Since then, sentiment analysis has matured into a sophisticated field, incorporating advanced NLP techniques and machine learning models to capture sentiments in real time across a wide array of digital media sources (Ramasangu, 2023).

2.2.1.1 Key Methodologies and Technologies in Sentiment Analysis

Sentiment analysis methodologies have evolved in tandem with advancements in NLP and AI technologies (M. Anwer, 2021). Early approaches relied on lexicon-based sentiment analysis, using predefined lists of words and phrases associated with positive and negative sentiments. While effective to some extent, these approaches struggled with context and sarcasm (Yadav, 2023).

Machine learning techniques, particularly supervised learning, revolutionized sentiment analysis by enabling models to learn sentiment patterns from labeled datasets. Techniques such as SVM, Naive Bayes NB, and more recently DL models like RNN and Transformers, have significantly improved sentiment analysis accuracy (Iparraguirre-Villanueva, 2023). Pre-trained language models, such as BERT and GPT-3, have further elevated sentiment analysis by capturing contextual nuances and domain-specific sentiment (Xu A. M., 2023).

2.3 Overview of Digital Media Content Analysis for Decision Making

Digital media content analysis is a crucial tool for decision-making in the rapidly evolving landscape of media and communication (Capatina, 2019). As an interdisciplinary field, it involves systematic examination and interpretation of various forms of digital media content, such as text, images, and videos, to derive meaningful insights. Scholars in this domain employ diverse methodologies, including natural language processing and machine learning algorithms, to analyze vast datasets and discern patterns, sentiments, and trends (Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, 2015). This literature review aims to provide an overview of the key methodologies and findings in digital media content analysis, emphasizing its role in

informing decision-making processes across industries. By synthesizing existing research, this review seeks to contribute to a comprehensive understanding of the applications, challenges, and potential future developments in leveraging digital media content analysis for effective decision-making.

2.3.1 Significance of Digital Media Content in Decision Making

This study (Hamad, 2016) addresses the growing use of Twitter in the healthcare domain and the need for a comprehensive research framework to analyze the content of health-related tweets. Examining 18 studies conducted between 2010 and 2014, the authors observed a lack of clear guidelines for combining quantitative and qualitative content analysis in Twitter-driven research. Consequently, they introduced the Combined Content Analysis (CCA) model, derived from key features of content analysis and mixed-methods research designs. The CCA model is proposed as a robust framework for designing, conducting, and evaluating investigations into Twitter-driven content in healthcare, aiming to enhance the methodological rigor and effectiveness of studies in this emerging field. The study underscores the importance of such a model and offers insights into its application, particularly in contexts related to elder care (Boumans, 2018).

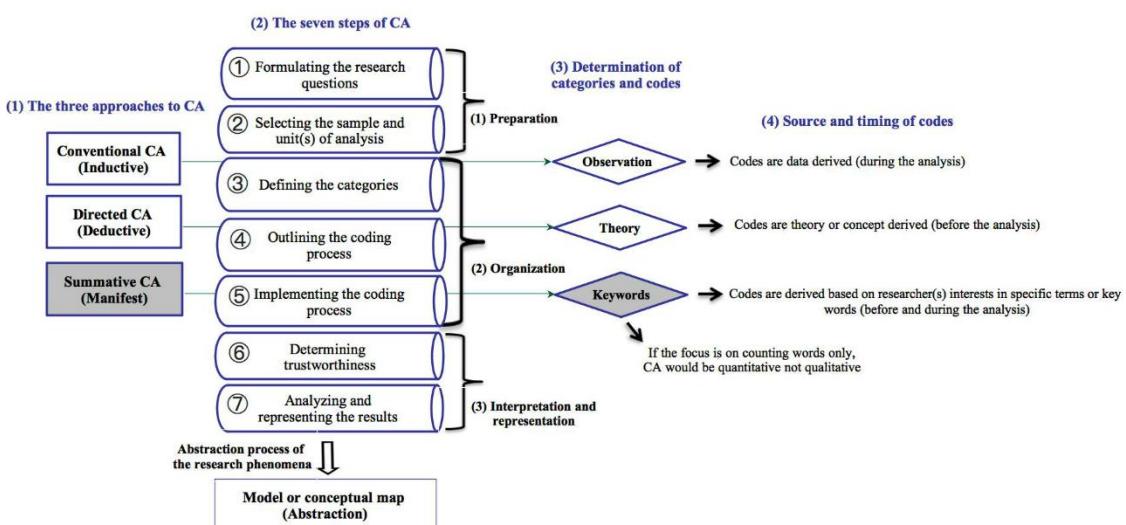


Figure 4 Content analysis in health care research.

2.3.2 Challenges and Opportunities in Analyzing Digital Media for Quality Management

While various studies have explored the potential of analyzing digital media for quality management, several challenges and opportunities have emerged in the existing literature (Li J. A., 2020). Challenges include the vast and dynamic nature of digital media content, making it challenging to collect, process, and analyze in real time. Additionally, ensuring the accuracy and reliability of sentiment analysis and opinion mining algorithms poses difficulties, as the contextual nuances of language can be complex to interpret accurately. Limited standardization in methodologies and tools for digital media content analysis further complicates comparisons across studies (Hackman, 1995).

Despite these challenges, opportunities abound in leveraging digital media for quality management. The wealth of unstructured data available on platforms such as social media provides an unprecedented source of customer feedback and opinions (Zonnenshain, 2020). Researchers have successfully employed natural language processing and machine learning techniques to extract meaningful insights from large datasets. Real-time monitoring of digital media allows for rapid identification of emerging trends and potential quality issues, enabling proactive decision-making (Ghermandi, 2019). Integrating artificial intelligence in quality management processes offers the potential for automation, predictive analytics, and continuous improvement (Stieglitz, 2018).

Noteworthy studies contributing to this understanding include research by (Naderi, 2023), who explored sentiment analysis in social media for business intelligence, and Liu (2015), who examined the challenges and opportunities of sentiment analysis in social media data. Additionally, (Chee, 2023) provided insights into using social media for quality improvement in healthcare. As the field continues to evolve, researchers must address these challenges and capitalize on the opportunities to harness the full potential of digital media for effective quality management.

2.4 The Evolution and Current State of Opinion Mining Techniques

The field of opinion mining has witnessed a significant evolution, progressing from early sentiment analysis approaches to more sophisticated techniques that incorporate natural language processing and machine learning (Binali, 2009). Initially focused on polarity identification, recent advancements encompass aspect-based sentiment analysis, emotion recognition, and context-aware opinion mining. Classical approaches often relied on rule-based systems, while contemporary methods leverage advanced algorithms, including deep learning models and ensemble techniques, to enhance accuracy and scalability (Messaoudi, 2022). Current research emphasizes the integration of multimodal data sources and the development of domain-specific models. The evolving landscape reflects a shift from binary sentiment classification to nuanced and context-aware opinion extraction, addressing the complexities of real-world language expressions. Key contributors to this evolution include (Tsui, 2023) and (Ning He). Understanding this trajectory is crucial for comprehending the state-of-the-art in opinion mining techniques and guiding future research in this dynamic and expanding field.

This paper (Rastogi, 2023) highlights the pivotal role of data mining in predicting diabetes, a significant global health concern. Utilizing techniques such as Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine on a real dataset from Kaggle, the study presents a diabetes prediction model implemented in Python. The logistic regression model exhibits the highest accuracy at 82.46%, outperforming other methods such as SVM. The research emphasizes the importance of early diabetes prediction for improved outcomes and proposes further exploration of additional classification algorithms to enhance prediction accuracy in future studies.

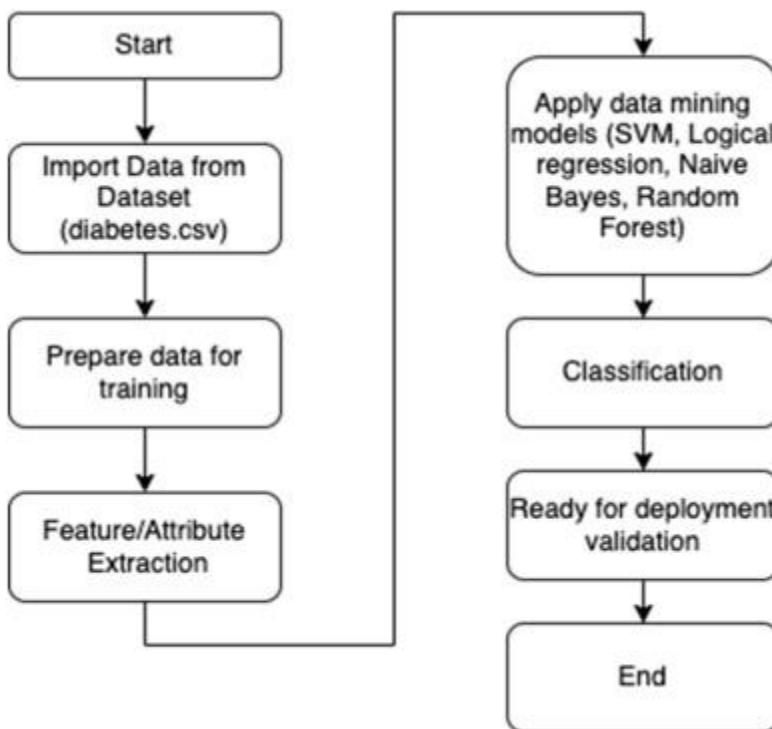


Figure 5 Data mining for diabetic data

The main finding of this (Guo, 2023) is the development and application of a novel framework, namely the wavelet fuzzy logic-based deep neural network (WFL-DNN), to enhance human resource management (HRM) systems in monitoring personnel performance and making informed decisions. The traditional HRM systems are highlighted as inadequate in handling the vast amount of data collected by businesses today. The proposed framework addresses this challenge by collecting raw employee data, preprocessing it through sampling, cleaning, and integration, and then utilizing principal component analysis (PCA) for feature extraction. Additionally, a genetic algorithm (GA) is employed for feature selection. The WFL-DNN framework is demonstrated to be more effective in accurately forecasting personnel performance compared to traditional methods. This finding suggests that incorporating advanced technologies, such as wavelet fuzzy logic and deep neural networks,

can significantly improve the precision and efficiency of HRM systems in evaluating and managing employee performance.

2.4.1 Traditional Approaches to Opinion Mining:

Early approaches to opinion mining primarily focused on rule-based systems and linguistic analysis to categorize sentiments expressed in text. Sentiment lexicons, predefined lists of words and phrases associated with positive or negative sentiments, were commonly employed for polarity classification. Additionally, rule-based algorithms utilized syntactic and semantic rules to identify sentiment patterns. Another traditional method involved machine learning techniques such as Support Vector Machines (SVM) and Naive Bayes for sentiment classification based on labeled training data. These approaches, while foundational, faced challenges in handling the complexity of language nuances and context.

The applications of sentiment analysis within digital media are multifaceted and extend across various domains:

1. **Content Creation and Optimization:** Content creators use sentiment analysis to gauge audience reactions to their articles, videos, or social media posts. Insights from sentiment analysis inform content optimization strategies, helping creators tailor content to audience preferences (Bibi, 2022).

2. **Engagement Strategies:** Organizations leverage sentiment analysis to identify viral content and trends. By understanding sentiment patterns, they can craft engagement strategies that resonate with their target audience and enhance brand loyalty (Leelawat, 2022).

3. **Quality Assurance:** Sentiment analysis plays a crucial role in maintaining content quality. It assists in content moderation by identifying inappropriate or harmful

content and helps fact-checkers and journalists identify misinformation and fake news (Neethu, 2013).

4. Advertising and Marketing: Marketers analyze sentiment to measure the effectiveness of advertising campaigns. They also use sentiment insights to personalize ad targeting and messaging.

5. News and Journalism: Sentiment analysis aids news outlets in understanding public sentiment towards news stories, political events, and social issues. This information can influence editorial decisions and story selection.

6. Public Opinion and Policy Making: Governments and policymakers monitor online sentiment to gauge public opinion on policy issues and to respond proactively to emerging trends or concerns (Alslaity, 2022).

The literature reviewed underscores the transformative potential of sentiment analysis in digital media [18]. It has evolved from a binary classification task to a sophisticated field empowered by AI and machine learning, offering valuable insights for decision-makers, content creators, and quality assurance processes (M. D. Nguyen, 2023).

2.4.2 Advancements and Innovations in Opinion Mining Using AI in digital media:

Recent advancements in opinion mining have been driven by the integration of artificial intelligence (AI) techniques. Machine learning algorithms, particularly deep learning models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs), have demonstrated superior performance in capturing intricate sentiment patterns and contextual information. Transfer learning has enabled models to leverage pre-trained language representations like BERT (Bidirectional Encoder Representations from Transformers) for more nuanced understanding of sentiments. Additionally (Ravi, "A survey

on opinion mining and sentiment analysis: tasks, approaches and applications., 2015), the fusion of natural language processing (NLP) with AI has facilitated aspect-based sentiment analysis, allowing systems to identify sentiments related to specific aspects or entities within a piece of text. These advancements not only enhance the accuracy of sentiment analysis but also enable the extraction of deeper insights from unstructured textual data. The incorporation of AI technologies represents a paradigm shift in opinion mining, offering more sophisticated tools for businesses to understand and respond to sentiments expressed in diverse sources such as social media, reviews, and forums.

The integration of Artificial Intelligence (AI) into journalism and communication practices has sparked profound debates about the transformative impact of technological innovations on society. This discourse echoes historical discussions, dating back to Marshall McLuhan's assertions in the 20th century that technologies extend human capabilities. Throughout the evolution of media technologies, from mechanical to electronic processes, they have been viewed as instruments of liberation and empowerment, expanding audience reach, overcoming spatial and temporal constraints, and bridging information gaps. With the advent of AI, a convergence of computing power with past innovations holds unprecedented potential for both positive and negative societal impacts. In the 21st century, network societies are more interconnected, transcending geographical boundaries and linking Westernized and global south societies. The proliferation of content creation by individuals and media organizations, facilitated by AI, has led to information overload, intensifying global and social challenges. Recognizing the exacerbated digital divide resulting from AI, this research proposes the Digital Dichotomy Theory (DD-Theory) as a framework to comprehend the inherent dynamics of global media communication. This theory aims to deepen our understanding of the complexities arising from AI's influence on communication

practices, offering insights into the evolving landscape of media in the digital age (Msughter, 2023).

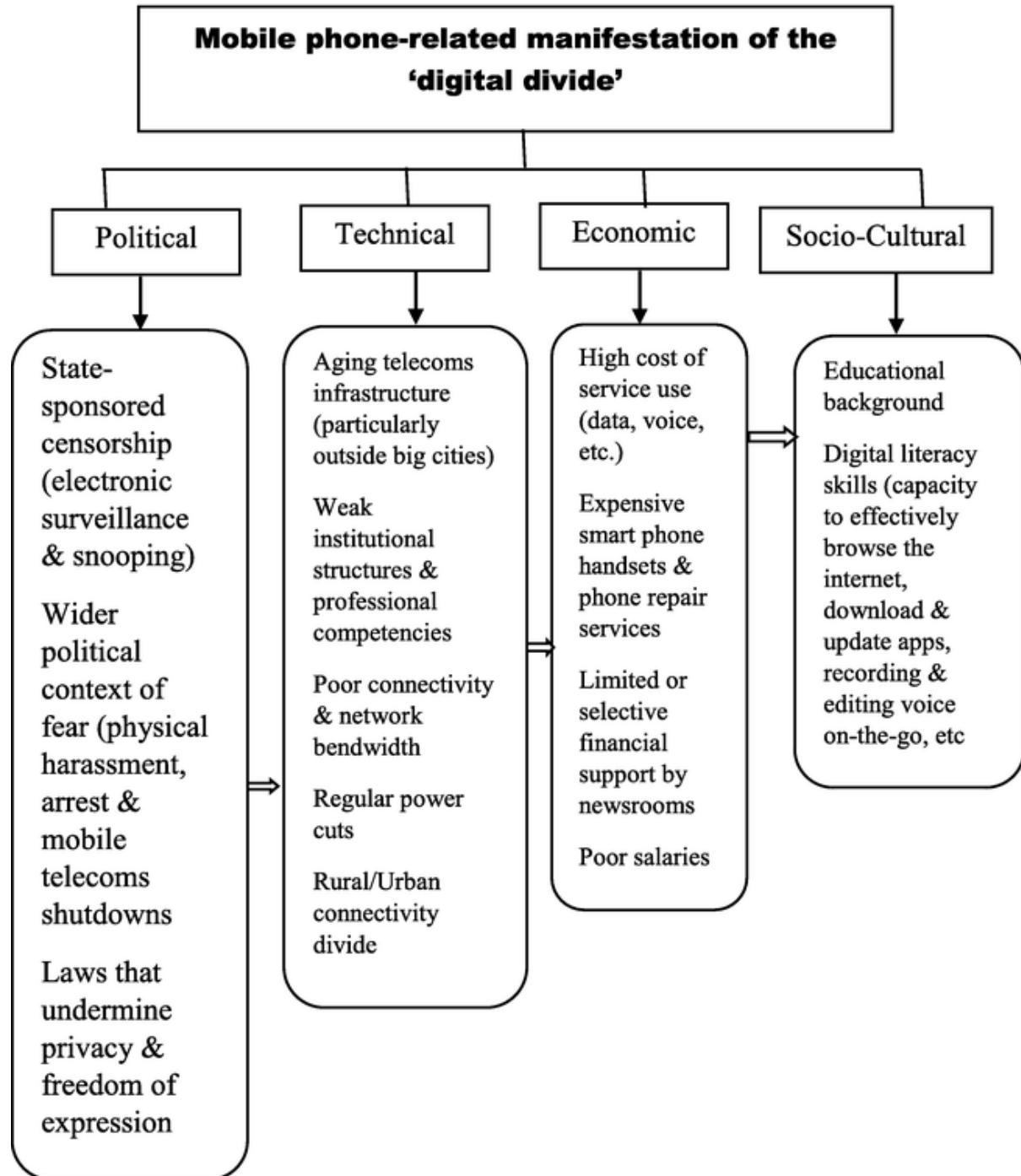


Figure 6 Mobile manifestation (Msughter, 2023).

2.5 Application of Opinion Mining in Various Industries using digital media

2.5.1 Opinion Mining in Retail and Consumer Goods:

In the retail and consumer goods industry, Opinion Mining plays a pivotal role in understanding customer sentiments and preferences (Kumar, 2021). By analyzing vast amounts of customer reviews, feedback, and social media comments, retailers can gain valuable insights into product satisfaction, identify emerging trends, and make data-driven decisions for inventory management and product development. Sentiment analysis helps in gauging consumer perceptions, improving customer experience, and tailoring marketing strategies. Additionally, tracking sentiment trends over time enables retailers to stay agile in responding to shifting consumer expectations and market dynamics (Ling, 1998).

In the landscape of small businesses, particularly startups, significant internal constraints such as limited financial capital and restricted access to quality human resources necessitate a cautious approach to adopting technologies that align with their business plans. Although digital marketing has emerged as a cornerstone for gaining a competitive edge in the 21st century, its efficacy with startups has encountered limitations. This (Mathur, 2023) study aims to evaluate the digital marketing practices employed by startups and delve into their strategies for creating brand awareness, fostering consumer loyalty, and deepening customer connections (Archak, 2007). The research methodology embraced a qualitative analysis, utilizing in-depth interviews with three selected startup businesses in India. Through this investigation, valuable managerial implications for startups in leveraging digital marketing techniques to achieve their marketing objectives have been unearthed. This research not only provides actionable insights for startups to optimize their digital marketing efforts but also establishes a foundation for future research endeavors on this crucial intersection of entrepreneurship and digital strategies (Castelo-Branco, 2020).

In the contemporary landscape of e-commerce, navigating through an extensive array of products to make well-informed purchasing decisions poses a considerable challenge for consumers. The need to compare prices, quantities, and features, such as the eco-friendliness of products, adds complexity to the decision-making process. Conversely, retailers face the challenge of identifying optimal replacements for out-of-stock items, catering to customer preferences. To address these challenges, this study employs a well-established methodology utilizing the Term Frequency-Inverse Document Frequency (TF-IDF) technique to represent products. Subsequently, K-means clustering is applied to group similar products, facilitating a structured organization of the product landscape. A significant challenge encountered is the poor quality of available textual data associated with the products. Despite this limitation, the results indicate the efficacy of the approach in successfully grouping genuinely similar products, offering valuable insights into their distribution. Notably, the study emphasizes that the analysis is solely based on product information, devoid of consumer data or purchase history, underlining the potential of this methodology to enhance product categorization and recommendations in e-commerce platforms (Srinita, 2023). This (Rudkowsky, 2018) literature review establishes the groundwork for understanding the implications and advancements in product clustering methodologies in the absence of consumer-centric data in the e-commerce domain.

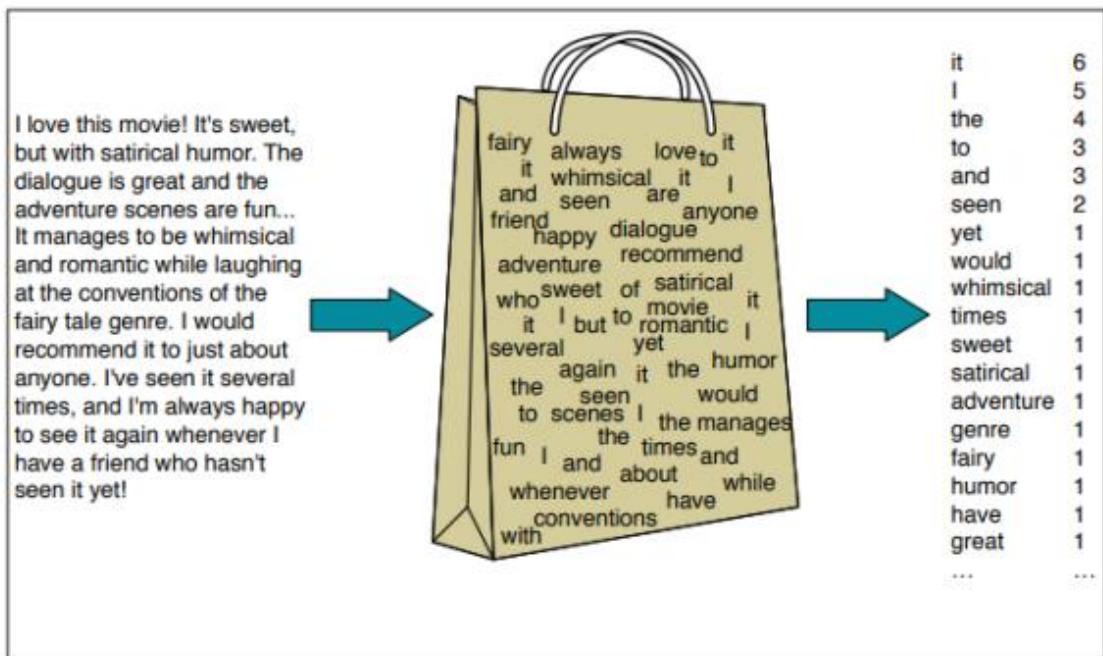


Figure 7 bag of words

2.5.2 Opinion Mining in Healthcare Quality Management:

In healthcare quality management, Opinion Mining contributes to the assessment of patient experiences, satisfaction levels, and feedback regarding medical services (Barry, 2017). Analyzing sentiments expressed in patient reviews and feedback forms aids healthcare providers in identifying areas for improvement, enhancing patient care, and optimizing overall service quality. Opinion Mining techniques allow healthcare organizations to proactively address concerns, prioritize quality initiatives, and maintain a patient-centric approach. Moreover, the continuous monitoring of opinions assists in benchmarking against industry standards and fostering a culture of patient-centered care.

This (Li R. Y.-Q.-Q.-N.-Y., 2023) study delves into the emotional heterogeneity among customers engaging in peer-to-peer accommodations, employing deep learning technologies and social network analysis to uncover key drivers. The findings shed light on the nuanced emotional responses within this sharing economy context. Notably, the environment emerges

as a fundamental driver of customer emotions in common scenarios, with services playing a less critical role in eliciting positive emotions. In specific situations, positive emotions are linked to factors such as price value and location, while the absence of these elements does not necessarily induce negative emotions. Surprisingly, the lack of household amenities does not impede the formation of positive emotions, revealing a unique aspect of customer emotional experiences in the peer-to-peer accommodation realm. Conversely, negative emotions are triggered by deficiencies in booking information. This study introduces a comprehensive understanding of the multifaceted demands and emotional causes shaping customer experiences in the sharing economy. The identified drivers offer valuable insights for tailored strategies to enhance customer emotional experiences in peer-to-peer accommodations, contributing to the literature on emotional heterogeneity and customer-centric strategies within the context of the sharing economy.

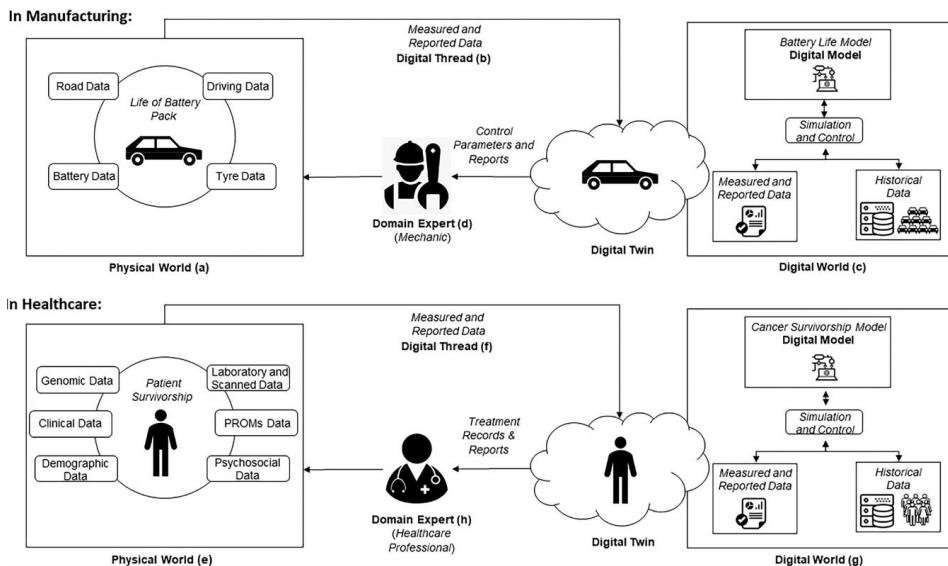


Figure 8 Digital twin

Opinion Mining in Service Industries (Hospitality, Finance, etc.): Opinion Mining is instrumental in service industries like hospitality and finance, where customer satisfaction is

paramount. In the hospitality sector, sentiments expressed in reviews on platforms like TripAdvisor or Yelp provide insights into the guest experience, helping businesses refine services, address concerns, and maintain high standards. In finance, Opinion Mining can analyze sentiments in social media and news articles to gauge market sentiment, anticipate trends, and inform investment decisions. Understanding customer opinions in these service-centric industries is crucial for reputation management, customer retention, and strategic planning, ultimately influencing business success and competitiveness.

2.6 Integration of AI-driven Opinion Mining in Quality Management:

The integration of Artificial Intelligence (AI)-driven Opinion Mining in quality management represents a transformative approach to enhancing product and service quality. By leveraging advanced sentiment analysis algorithms, natural language processing, and machine learning techniques, organizations can extract valuable insights from customer opinions, feedback, and reviews in real-time. This integration not only enables the identification of areas for improvement but also allows for the proactive management of quality-related issues. AI-driven Opinion Mining contributes to a more comprehensive understanding of customer sentiments, providing a holistic view of product and service performance.

2.6.1 Improving Product and Service Quality Using Opinion Mining Insights:

Opinion Mining insights derived from AI technologies play a pivotal role in improving product and service quality. By analyzing customer opinions, organizations can pinpoint specific features, attributes, or aspects that contribute to positive or negative perceptions. This information becomes invaluable for targeted quality enhancements, product innovation, and tailored service improvements. Real-time access to these insights empowers organizations to

address emerging issues promptly, fostering a continuous improvement culture that is responsive to customer needs and expectations.

2.6.2 Leveraging AI for Real-time Decision Making in Quality Management:

The integration of AI in Opinion Mining not only aids in understanding customer sentiments but also facilitates real-time decision-making in quality management. AI algorithms process large volumes of unstructured data, providing actionable insights that guide immediate decisions to uphold or enhance product and service quality. This proactive approach to decision-making allows organizations to swiftly address quality concerns, implement corrective measures, and optimize processes for better outcomes. The real-time nature of AI-driven Opinion Mining ensures that quality management becomes an agile and dynamic process, responsive to the evolving landscape of customer preferences and industry standards.

2.7 Gaps in the literature

While existing literature has made significant strides in exploring the integration of AI-driven Opinion Mining in quality management, certain research gaps warrant attention. One notable gap lies in the need for a deeper understanding of the contextual factors influencing the effectiveness of Opinion Mining models in diverse industries. Prior studies often focus on specific sectors, and there is a lack of comprehensive frameworks that can be applied universally. Additionally, the impact of cultural nuances on the interpretation of opinions remains underexplored. Studies may benefit from investigating how cultural variations influence the perception of quality and the performance of Opinion Mining algorithms. Moreover, there is a dearth of research on the ethical considerations associated with the use of AI in Opinion Mining for quality management, highlighting the need for

studies addressing privacy concerns, algorithmic bias, and transparency in decision-making processes.

In our research, notable gaps emerge, urging further exploration in the realm of sentiment analysis in digital media (Mann, 2022). These include the integration of multi-modal data sources, the development of real-time decision support systems, cross-platform sentiment analysis, ethical considerations, and user-centric sentiment analysis (Kaur, 2023). Closing these gaps promises to enhance the depth and breadth of sentiment analysis applications, ensuring its ethical use, and fostering personalized, real-time decision-making in the dynamic digital media landscape (Shrivastav, 2023).

Table 2 Gap Analysis from previous research

Paper	Technique	Dataset	Accuracy Achieved	Application	Pros	Cons
(Mar djo, 2022)	Approaches Using Machine Learning and Lexicons	Twitter dataset	83.3%	Sentiment analysis of tweets	Machine learning can analyze text without feature engineering	Traditional methods require feature engineering
(Joloudari, 2019)	Long Term Memory (LSTM) and Convolutional Neural Networks (CNN)	English language tweets	88.5%	Sentiment analysis of tweets	Efficient and reliable technique	-
(Hasan, 2019)	Convolutional, recurrent, neural networks,	-	-	Sentiment analysis of texts	Can recognize new	Less accurate than supervise

	unsupervised , and mixed neural networks, as well as deep reinforcement learning				complex features	deep techniques
(Shamsrat, 2021)	Algorithms for supervised machine learning, such as Support Vector Machines and Artificial Neural Networks	User-created texts	88.5%	Sentiment analysis of texts	Can extract users' feelings from their writing	Slow and take a long time to train
(Ahmad, 2017)	Support vector machines (SVM) and Universal Language Model Fine-tuning (ULMFiT)	-	-	Sentiment analysis of texts	Powerful deep learning architecture	-
(Aslan, 2023)	ELMO and CNN	Twitter dataset	-	Clustering in service discovery	Effective discovery of the best service	-

2.8 The research questions

1. Why is Sentiment Analysis Important, and How Can We Improve Existing Methods?

- In the context of decision-making for quality management in digital media content, sentiment analysis is crucial for gauging user reactions, preferences, and feedback. Previous research, such as the work by (Alslaity, 2022) and (Bibi, 2022) emphasizes the significance of sentiment analysis in understanding user sentiments and its applications in diverse industries. The research seeks to build upon existing methodologies, exploring improvements and innovations in sentiment analysis techniques. Leveraging insights from studies like (al, 2020), the goal is to surpass current methods, enhancing accuracy and relevance for effective decision-making in quality management.

2. Data Collection: Where and How?

- Previous studies, such as the work by (Elmitwally, 2020) on sentiment analysis in multimedia content, shed light on the challenges and opportunities in collecting relevant data for sentiment analysis. Building on these insights, the research will explore diverse sources, potentially incorporating methodologies from (O. Araque, 2017) on mining social media data for sentiment analysis. The goal is to identify pertinent sources and employ optimized methods for comprehensive data collection aligned with the objectives of quality management.

3. Conversion of Multimedia User Reviews to Text: Methodology and Challenges

- The transformation of multimedia user reviews into text is a critical step. Previous works, like that of (Elmitwally, 2020) on multimodal sentiment analysis, provide a foundation for understanding challenges in processing diverse content forms. The research aims to draw from these insights to address challenges in the conversion process, potentially incorporating

methodologies from studies by (Chee, 2023) on audio-visual sentiment analysis.

4. Sentiment Analysis on Text: Methodologies and Techniques

- The methodology for sentiment analysis on textual data will be influenced by previous works such as the study by (Kabiri, 2019) on recursive deep models for sentiment analysis. Building upon these techniques, the research seeks to explore advanced methodologies, potentially incorporating insights from recent developments in natural language processing, deep learning, and machine learning, as seen in works by Devlin et al. (2018) and Vaswani et al. (2017).

5. In-Depth Labeling and Classification of Sentiments: Post Sentiment Analysis

- In-depth sentiment labeling and classification are critical for actionable insights. Prior research, including that by Turney (2022) on unsupervised sentiment classification and (E. Cambria, 2020) on a sentimental education, provides foundational concepts. The research aims to extend these ideas by incorporating advanced techniques such as those discussed in studies like Kim (2014) on convolutional neural networks for sentence classification. This ensures a nuanced and comprehensive classification system aligned with the objectives of quality management decision-making.

2.9 Opportunities for contributions to knowledge

The research presents promising opportunities for advancing knowledge in sentiment analysis, opinion mining, and quality management within the realm of digital media content. By innovating sentiment analysis methodologies, particularly in handling textual data derived

from diverse multimedia sources, the study aims to contribute to the evolution of techniques for more accurate and efficient sentiment analysis. The opportunity to enhance sentiment labeling and classification systems using advanced machine learning and natural language processing techniques is pivotal, offering the potential for nuanced and granular outcomes. Additionally, the integration of artificial intelligence into quality management decision-making processes opens avenues for actionable insights, contributing to the broader discourse on responsible AI applications. Addressing ethical considerations and privacy concerns within the context of sentiment analysis and opinion mining, along with exploring industry-specific adaptations, further positions the research to make valuable contributions to the evolving landscape of sentiment analysis and quality management in the digital era.

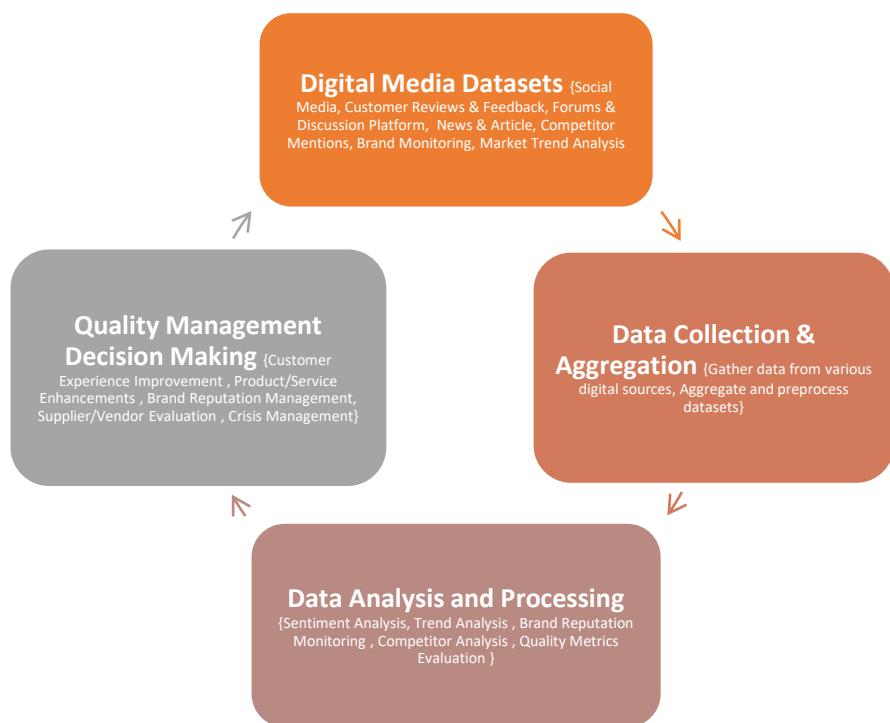


Figure 9 Opportunities framework for contributions to knowledge

- Digital media datasets, comprising various sources like social media, customer reviews, forums, news, and competitor mentions, form the primary input.

- Data collection involves aggregating and preprocessing these diverse datasets to make them suitable for analysis.
- The analysis phase involves techniques such as sentiment analysis, trend analysis, brand reputation monitoring, competitor analysis, and evaluation of quality metrics.
- The insights derived from the analysis phase contribute to informed decision-making in Quality Management. This includes improving customer experience, enhancing products/services, managing brand reputation, evaluating suppliers/vendors, and addressing crises or quality-related issues.

2.10 Conclusions

The comprehensive exploration conducted in this literature review delves into the multifaceted domain of opinion mining and its pivotal role in the landscape of Quality Management Decision Making through the prism of artificial intelligence (AI) methodologies. The critical analysis undertaken has unveiled a rich tapestry of existing knowledge, elucidating the foundational elements requisite for the implementation of a pioneering opinion mining approach in analyzing digital media content. By synthesizing and assimilating key research domains, this review has substantiated the significance of AI-driven opinion mining as a transformative tool poised to elevate quality management strategies across diverse industries. It underscores the contextual underpinnings and accentuates the symbiotic relationship between AI advancements and the burgeoning terrain of digital media content analysis. This synthesis illuminate's pathways for future endeavors, advocating a paradigm shift towards harnessing AI's potential for refining decision-making processes and augmenting quality assurance efforts within the ever-evolving digital landscape.

Chapter III.

Methodology

A summary of the methodological decisions, research viewpoint, and procedures used to address the research questions is given in this chapter. Following the introduction of the thesis research methodology, the explanation of the methodical literature review technique and the selection of both quantitative and qualitative methodologies, the chapter addresses the methodology of research encompassing various sources of data & statistical techniques.

3.1 Introduction

The research methodology employed for implementing a novel opinion mining approach on digital media content using artificial intelligence (AI) for decision-making and quality analysis is a balanced integration of qualitative and quantitative techniques. This methodological fusion aims to comprehensively explore the subjective nuances of opinions through qualitative analyses like content analysis, interviews, and case studies, while also employing AI-driven quantitative methods such as sentiment analysis, machine learning algorithms, and statistical analysis to numerically assess and categorize large volumes of data. This mixed-methods approach facilitates a holistic understanding of decision-making dynamics and content quality assessment by leveraging insights from both subjective expressions and objective data patterns. Such an approach, widely acknowledged and valued across diverse research domains, ensures robustness in findings by triangulating multifaceted perspectives, enhancing the reliability and validity of the research outcomes. This methodology framework not only delineates the systematic approach to exploring opinion mining in digital media content but also sets the stage for a comprehensive understanding of

its applications and implications within this domain.

The methods chapter serves as the blueprint of the research design, focusing on ensuring rigor, ethical considerations, and justifying the choices made in the research process. Rigor in research design involves meticulous planning and execution to produce reliable and valid results. Ethical considerations are paramount, ensuring that the research respects the rights and well-being of participants and adheres to ethical guidelines. Justification plays a pivotal role in outlining the rationale behind the decisions made in the research methodology.

The research design is structured to uphold rigor through a carefully crafted integration of qualitative and quantitative techniques. Qualitative methods, such as content analysis, interviews, and case studies, delve into the subjective aspects of opinions expressed in digital media content. These techniques allow for in-depth exploration and understanding of nuanced perspectives, crucial for contextualizing the findings within the research area.

On the other hand, quantitative methods, including sentiment analysis, machine learning algorithms, and statistical analysis, provide a numerical assessment of data derived from digital media content. These techniques enable the analysis of large datasets to identify patterns, trends, and sentiment polarities, augmenting the comprehensive understanding of the subject matter.

Ethics and reasoning are pivotal components guiding the research process. Ethical considerations are prioritized throughout, ensuring the protection of participants' rights, confidentiality, and informed consent in data collection methods. Reasoning underpins every decision, emphasizing the critical analysis of choices made in the research design, data collection, and analysis methods. Transparency in reasoning allows for a clear explanation of why certain methods were chosen over others, providing a robust foundation for the research methodology.

Every decision made in the research design is meticulously justified, outlining the 'why' behind each chosen method or approach. This constant emphasis on justification ensures that the research design aligns with the research objectives, enabling a comprehensive and methodologically sound exploration of the novel opinion mining approach on digital media content using AI for decision-making and quality analysis. Ultimately, the methods chapter not only describes the chosen methodologies but also serves as a testament to the rigor, ethical integrity, and reasoning behind the research design.

3.2 Research Design: Structuring Rigorous Investigations

In the pursuit of developing a comprehensive understanding of sentiment analysis within the context of digital media content, insights gleaned from previously published research have been integrated. Notably, "A Novel Approach to Predict the Real-Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE" and "Sentiment Analysis Predictions in Digital Media Content Using NLP Techniques" have provided foundational insights into the methodologies and approaches adopted in sentiment analysis amidst real-time events and digital content.

The methodology employed in the first paper centered on leveraging Naive Bayes and Recurrent Neural Network (RNN) algorithms for real-time sentiment analysis during the COVID-19 pandemic in the UAE. This approach provided valuable insights into handling real-time data streams and modeling sentiment shifts in response to significant events.

In the second paper, the focus was on utilizing Natural Language Processing (NLP) techniques for sentiment analysis predictions within digital media content. The

methodologies outlined in this research shed light on the use of linguistic analysis, feature extraction, and sentiment classification in the context of diverse digital content sources.

The frameworks, algorithms, and data preprocessing techniques elucidated in these papers have directly influenced the design and selection of methodologies for the present study. Specifically, the adoption of similar algorithms, preprocessing steps, and feature extraction methods has been guided by the successes and learnings derived from these previous works.

In the current study, the adaptation and refinement of these methodologies have been tailored to suit the specific nuances of sentiment analysis in digital media content. Modifications and enhancements have been made in consideration of the unique characteristics of the data sources and the research objectives.

3.2.1 Insights from Previous Research Papers

3.2.1.1 "A Novel Approach to Predict the Real-Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE"

At present, each and every part of the globe are facing a COVID crisis, which is affecting an individual physically, mentally and on the other hand, it is affecting the nation economically. Also, the unemployment scenario will be at its peak in the upcoming years as reported by UNGA. To combat this scenario, all the country are working on fostering their fiber network and so the sectors apart from the manufacturing, will tend to work from their home and contribute to the economy. But there are many problems arising to implement this culture practically, since it affects the mindsets of the people who have to endure this transformation within a very short span. Hence, in this research work, it has been decided to focus on this current issue for which the usage of certain apps in UAE such as zoom, totok, botim for internet calling have been identified since this is the only way of connectivity with the outside world. To perform this analysis, the tweets from December to July have been

collected by converting the image to text and analyzed using two algorithms such as Naive Bayes Classifier (NBC) and Recurrent Neural Networks (RNN). The sentimental analysis found that 630 tweets were positive and people in UAE feels secured, satisfied and internet calling is very useful for them in the prospect of work, education, etc. Only 48 tweets has negative impact because people feel little bit harder in sudden change of culture with in short period of time and 155 tweets has impact that both positive and negative were view and said to be natural. The study found that NB (84%) is more accurate, user friendly and takes less time than RNN (79%) to perform the analysis. Finally, the sentimental analysis reveals that people in UAE were accepting the new culture of internet calling and it is useful for them in the prospect of work and education

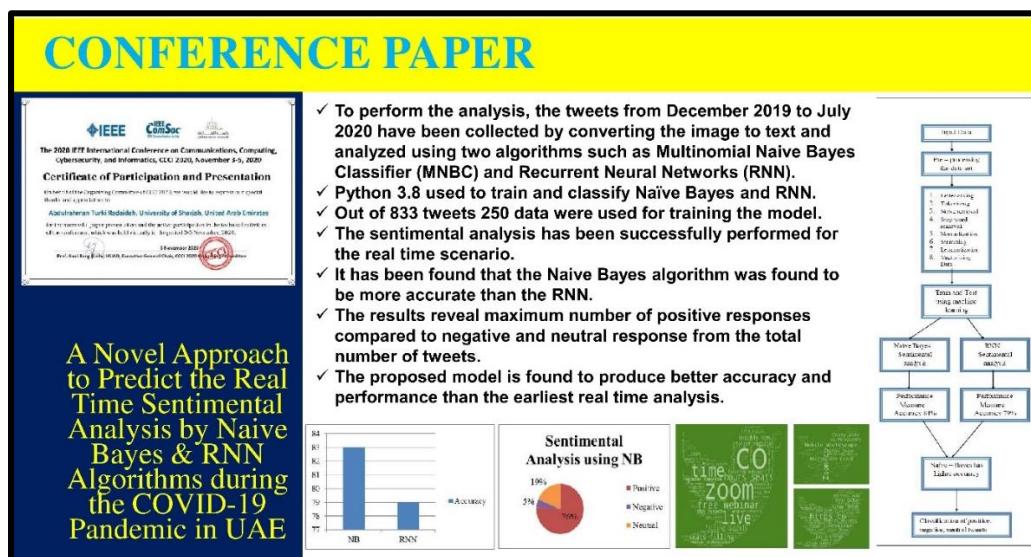


Figure 10 conference paper detail

3.2.1.1 Methodologies Employed: Naive Bayes and RNN Algorithms for Real-Time Sentiment Analysis

Under this section, we used dataset from twitter and analyzed it. The model used in the study was NBC and RNN along with clustering of sentimental analysis. The accuracy of the both the method NBC and RNN will be collected and the method which as

high accuracy will cluster to find the factors for the sentimental analysis. Using the datasets collected, the training will be done and the corresponding output will be generated.

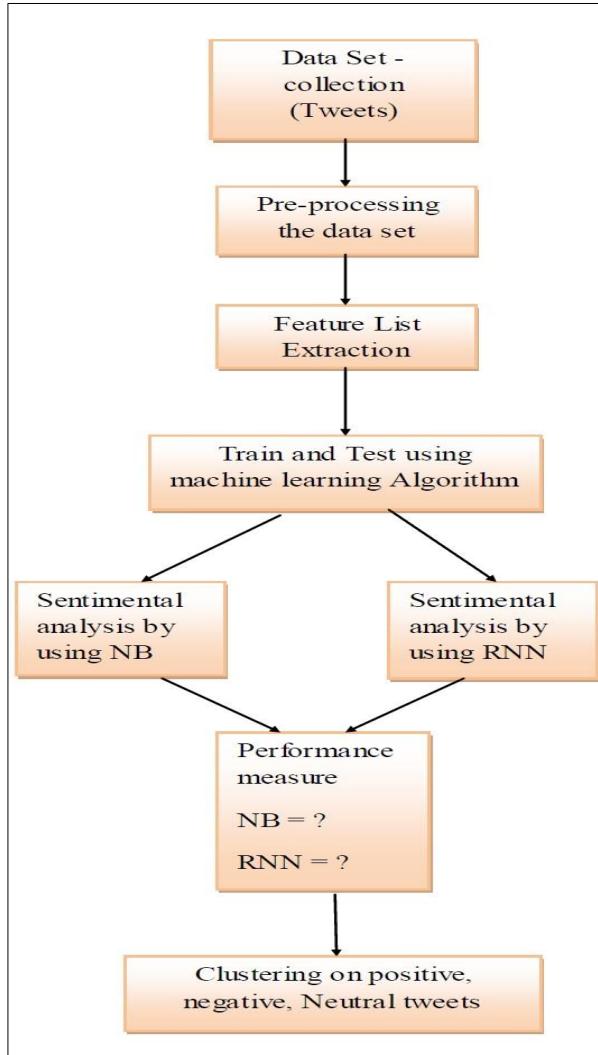


Figure 11 Research Procedure for Naive Bayes and RNN Algorithms for Real-Time Sentiment Analysis

A. Data Pre-processing

It is the first step for data analysis for which the various data has been collected from the twitter from December to July after the onset of COVID pandemic and it has been classified with reference to internet calling. This is an important step because the quality of the data will lead to more reliable results. Preprocessing a Twitter dataset involves a series of

tasks like removing all types of irrelevant information like emojis, special characters, and extra blank spaces. It can also involve making format improvements, delete duplicate tweets, or tweets that are shorter than three characters. There are eight steps for Data Pre – Processing Tweets. They are:

1. Letter casing: Converting all letters to either upper case or lower case.
2. Tokenizing: Turning the tweets into tokens. Tokens are words separated by spaces in a text.
3. Noise removal: Eliminating unwanted characters, such as HTML tags, punctuation marks, special characters, white spaces etc.
4. Stop word removal: Some words do not contribute much to the machine learning model, so it's good to remove them. A list of stop words can be defined by the nltk library, or it can be business-specific.
5. Normalization: Normalization generally refers to a series of related tasks meant to put all text on the same level. Converting text to lower case, removing special characters, and removing stopwords will remove basic inconsistencies. Normalization improves text matching.
6. Stemming: Eliminating affixes (circumfixes, suffixes, prefixes, infixes) from a word in order to obtain a word stem. Porter Stemmer is the most widely used technique because it is very fast. Generally, stemming chops off end of the word, and mostly it works fine. Example: Working -> Work
7. Lemmatization: The goal is same as with stemming, but stemming a word sometimes loses the actual meaning of the word. Lemmatization usually refers to doing things properly using vocabulary and morphological analysis of words. It returns the base or dictionary form of a word, also known as the lemma .Example: Better -> Good.

8. Vectorizing Data: Vectorizing is the process to convert tokens to numbers. It is an important step because the machine learning algorithm works with numbers and not text.

B. Training and Classification

Supervised learning is an important technique for classification problems. In the study we used two supervised tool for sentimental analysis. The two supervised tools said to be Naive Bayes and recurrent neural networks.

3.2.1.1.2 Influence on Current Study: Adapting Real-Time Analysis Techniques

We used Python to train and classify Naïve Bayes and RNN. Out of 833 tweets 250 data were used for training the model. The Fig shows the overall flow of the process

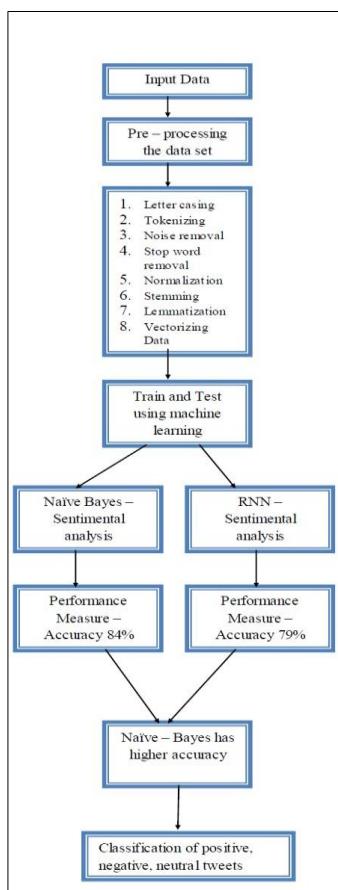


Figure 12 Proposed Framework: A Novel Approach to Predict the Real-Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE

Table 3 Performance Measure of NB

Table - 1 (NB)	
Performance Measure	
Positive Recall	81.3
Negative Recall	66.06
Positive Precision	84.5
Negative Precision	62.69

Table 4 Performance Measure of RNN

Table - 2 (RNN)	
Performance Measure	
Positive Recall	77.42
Negative Recall	68.23
Positive Precision	79.82
Negative Precision	65.32

Table 5 Accuracy of methods

Table - 3	
Methods	
Methods	Accuracy

NB		83
RNN		79

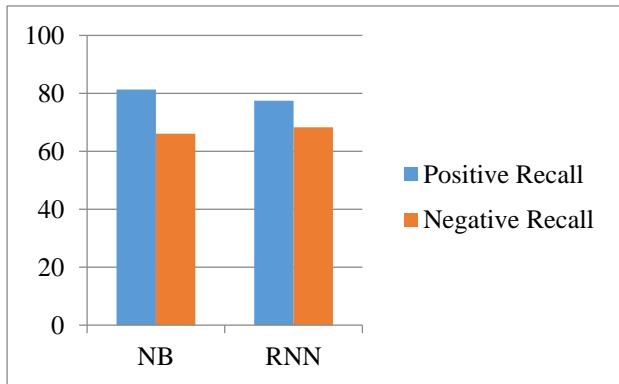


Figure 13 Measurement of positive and negative recall of the technique

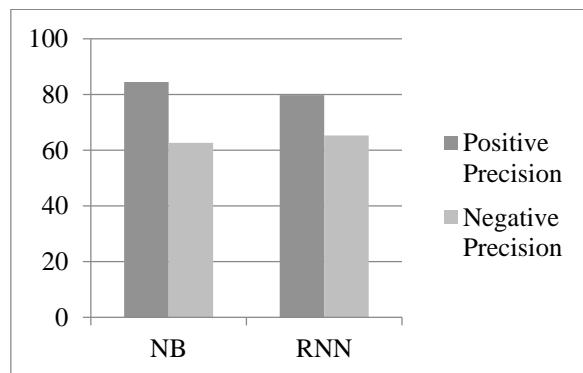


Figure 14 Measurement of positive and negative precision of the technique

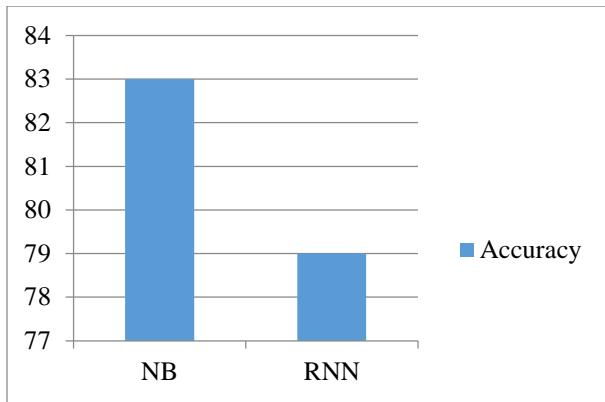


Figure 15 Accuracy of the methods RNN and NB

From the above Table 1 and Table 2 shows the performance measure of Naive Bayes, RNN in the terms of Recall and Precision. Along with that, Table 3 shows the performance classifier terms of accuracy. Likewise from the Figure 3 and Figure 4 shows the positive, negative recall and positive, negative precision respectively. Figure 5 shows the Accuracy of the methods RNN and NB. The accuracy of Naive Bayes (NB) was greater than RNN, hence the positive, negative and neutral analysis of NB shown in Fig.

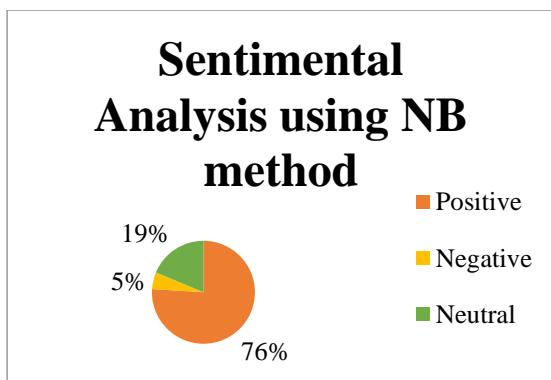


Figure 16 – NB Sentiment breakdown

From the analysis, it has been found that the usage of Internet calling was found to be predominant for performing the day-to-day operations in this pandemic scenario. In addition, the results reveal that the maximum number of positive responses has been recorded followed by negative and neutral response (Fig.6) from the tweets and on the other hand, it can be concluded that this culture is mostly agreed by most of the individuals in UAE.

Clustering

On clustering the positive, negative and neutral analysis it is found that people in UAE feels secured, satisfied and internet calling is very useful for them in the prospect of work, education, etc. Negative analysis shows that people were struggle to adapt to the situation and neutral analysis were people how have the capable to adopt to the culture.

3.2.1.2 "Sentiment Analysis Predictions in Digital Media Content Using NLP Techniques"

In the current digital landscape, understanding sentiment in digital media is crucial for informed decision-making and content quality. The primary objective is to improve decision-making processes and enhance content quality within this dynamic environment. To achieve this, a comprehensive comparative analysis of NLP for tweet sentiment analysis was conducted, revealing compelling insights. The BERT pre-trained model stood out, achieving an accuracy rate of 94.56%, emphasizing the effectiveness of transfer learning in text classification. Among machine learning algorithms, the Random Forest model excelled with an accuracy rate of 70.82%, while the K Nearest Neighbours model trailed at 55.36%. Additionally, the LSTM model demonstrated excellence in Recall, Precision, and F1 metrics, recording values of 81.12%, 82.32%, and 80.12%, respectively. Future research directions include optimizing model architecture, exploring alternative deep learning approaches, and expanding datasets for improved generalizability. While valuable insights are provided by our study, it is important to acknowledge its limitations, including a Twitter-centric focus,

constrained model comparisons, and binary sentiment analysis. These constraints highlight opportunities for more nuanced and

3.2.1.2.1 Methodologies Utilized: NLP Techniques for Digital Media Content Analysis

The study utilized a dataset obtained from the Kaggle website, comprising around 160,000 tweets from the Twitter blog categorized into three groups: positive, negative, and neutral. Initial analysis involved applying various pre-processing techniques to cleanse and prepare the tweets for feature extraction. Subsequently (Madhu, 2023), the database was split into a training set and a test set. Features were extracted from tweets using diverse techniques, and machine learning algorithms were trained for tweet polarity classification, including support vector machine, naive Bayes, decision tree, and K-nearest neighbor approaches (Sufi, 2022). The performance of these classifiers was then assessed on the test set across all extraction techniques to compare their impact on the sentiment analysis process, using multiple performance evaluation measures. Following this evaluation, a model was proposed, specifically a recurrent neural network employing Long Short-Term Memory (LSTM). The results obtained from this model were then compared with the outcomes of the previous classifiers (Fanni, 2023).

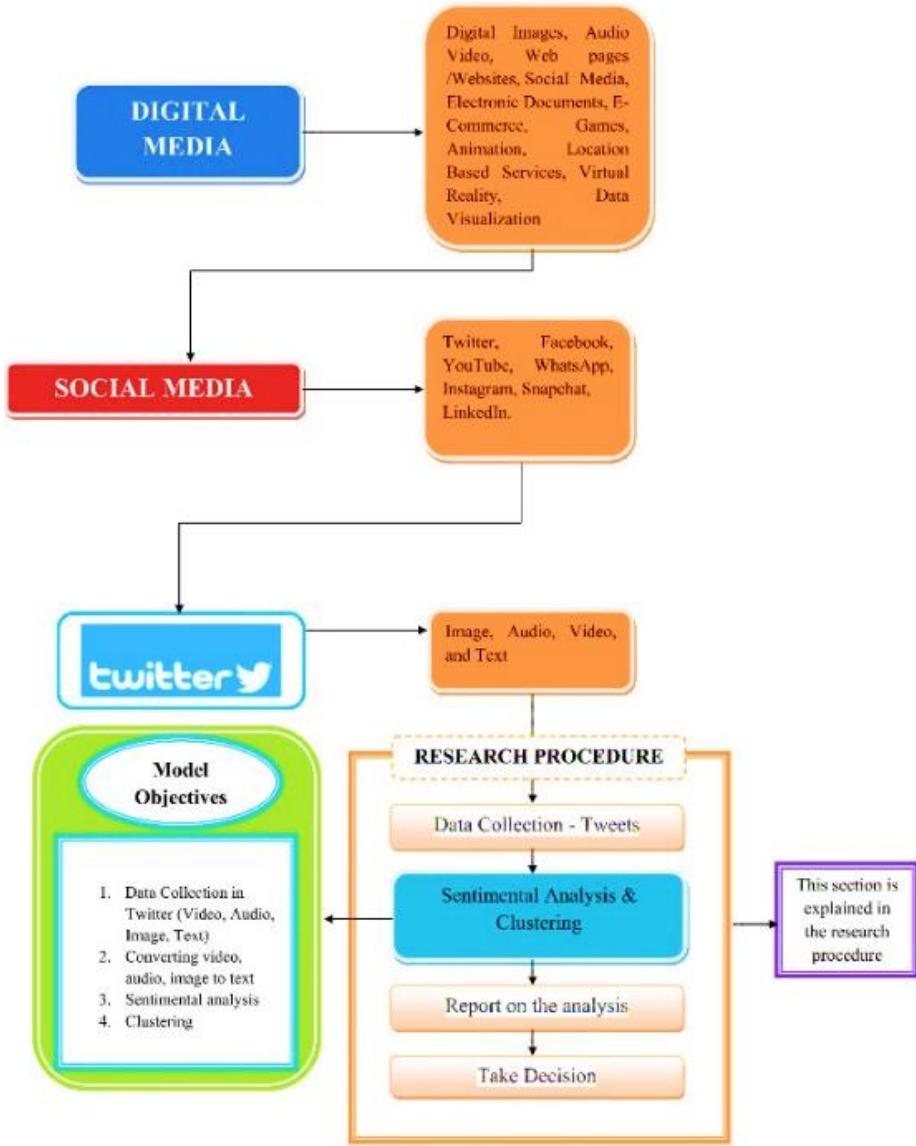


Figure 17 General Framework Sentiment Analysis Predictions in Digital Media Content Using NLP Technique

Tools and Resources

To accomplish the task of sentiment analysis, various tools and resources are necessary. The required tools and resources for analyzing the sentiment of tweets are discussed below:

- Programming Language:

Python, a high-level general-purpose programming language, proves to be an excellent tool for artificial intelligence, machine learning, and deep learning. Python will be used for creating and training models. Several Python libraries will be employed for sentiment analysis, including Pandas, Numpy, Scikit Learn, NLTK, Re, Keras, PyTorch, and Transformers.

Table 6 Python Libraries

ML Libraries	Function
From sklearn.metrics import accuracy_score:	find the accuracy score
from sklearn.naive_bayes import MultinomialNB:	The term Multinomial Naive Bayes simply lets us know that each $p(f_i c)$ is a multinomial distribution, rather than some other distribution. This works well for data which can easily be turned into counts, such as word counts in text.
from sklearn.linear_model import Logistic Regression:	Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. Mathematically, a logistic regression model predicts $P(Y=1)$ as a function of X .
from sklearn.svm import SVC:	The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is
import numpy as np:	NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.
from nltk.corpus import stopwords:	Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. ... Such words are already captured this in corpus named corpus
from nltk.tokenize import word_tokenize:	We are able to extract the tokens from string of characters by using tokenize.word_tokenize() method.

	<p>It actually returns the syllables from a single word. A single word can contain one or two syllables.</p>
from sklearn.feature_extraction.text import TfidfVectorizer:	TfidfVectorizer - Transforms text to feature vectors that can be used as input to estimator. vocabulary_ Is a dictionary that converts each token (word) to feature index in the matrix, each unique token gets a feature index. ... In each vector the numbers (weights) represent features tf-idf score.
from sklearn.model_selection import train_test_split:	splitting the data to train and test
from nltk.stem import PorterStemmer:	The Porter stemming algorithm (or 'Porter stemmer') is a process for removing the commoner morphological and inflectional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.
from nltk.stem import WordNetLemmatizer:	Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .
import seaborn as sns:	Seaborn is a data visualization library in Python based on matplotlib. The seaborn website has some very helpful documentation, including a tutorial. ... Then simply include import seaborn as sns at the top of your python file
from tweepy import OAuthHandler:	User owns the app / Single User - If the user is the owner of the app, they can generate access tokens on the "Keys and Tokens" tab in an app's "Details" section within the Twitter app dashboard. Click the "Create" button in the "Access token & access token secret" section.
from tweepy.streaming import StreamListener:	Tweepy makes it easier to use the twitter streaming api by handling authentication, connection, creating and destroying the session, reading incoming messages, and partially routing messages.
import tweepy:	library to scrap the tweets
import json:	to read the json (java script object notation) data structure
import pandas as pd:	pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
import csv:	to read/write csv files
import re:	regular expression to work out on pattern matching

from textblob import TextBlob:	TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
import os:	to work with operating systemm commands
import time:	from datetime import datetime, that is to work with the date and time

- Software:

Anaconda, a Python distribution platform, will be utilized, providing access to many built-in packages. Within Anaconda, Jupyter Notebook will serve as the primary environment for developing and training machine learning and deep learning models.

- Twitter API:

After developing and evaluating the machine learning and deep learning models, the Twitter API will be employed to extract new tweets and test the model's performance. Twitter allows the use of 3rd party Python packages like Tweepy to extract tweets based on query words and date range, facilitating the entire process.

- Hardware:

Given that various machine learning and deep learning models will be trained, the minimum system requirements are as follows: Core i5 Processor, 16 GB of RAM, Nvidia GPU with a minimum of 6 GB of V-RAM, and 100 GB of HDD space.

Machine learning-Based Models

Through this model, we will initially process the data, extract the features, and then classify them based on machine learning algorithms. Figure (2) shows the work steps.

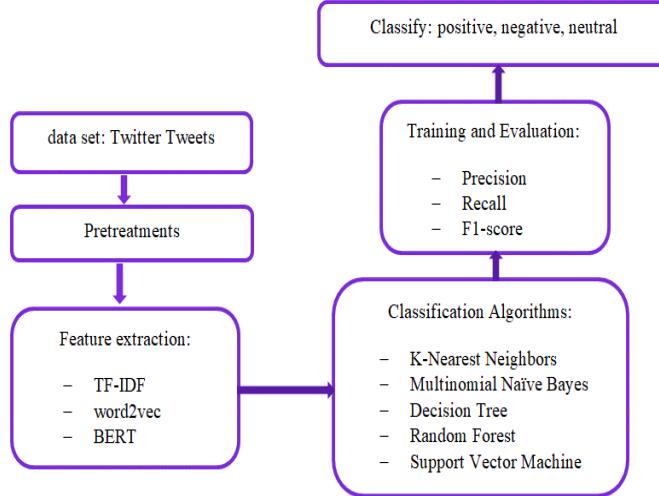


Figure 18 Framework of Model

The model involves several steps, including pre-processing the data, feature extraction using techniques such as TF-IDF, word2vec, and BERT, and classifying the tweets using various algorithms such as k-nearest neighbors, multinomial Naïve Bayes, decision tree, random forest, and support vector machine. The model will be trained and evaluated using performance metrics such as precision, recall, and F1-score.

Features Selection

1. Data pre-processing

Since machines do not understand spoken or written natural language, data pre-processing is a very important step for sentiment analysis and a necessary process before training machine learning models. Data pre-processing aims to make it easier to train and test classifiers by performing an appropriate set of transformations on the data. We did the pre-processing using the NLTK library in Python. Because we work with text data from Twitter. Tweets contain different parts that are not necessary or important to understanding the meaning of a tweet. Where we can extract the semantic meaning of the tweet by getting rid of all the unnecessary words and symbols by pre-processing the data. The tweets were pre-

processed by selecting features that were likely to be relevant for sentiment analysis. In general, when selecting features for sentiment analysis, some common criteria researchers consider include relevance, in-formativeness, redundancy, computational efficiency, and interpretability. The pre-processing of the data follows the following steps: remove punctuation, remove stop words, remove URLs, remove emoji, remove hash marks, and drop all word wrapping, derivation, and markup. The data is now clean and ready for feature extraction. Figure (3) shows the steps for data pre-processing.

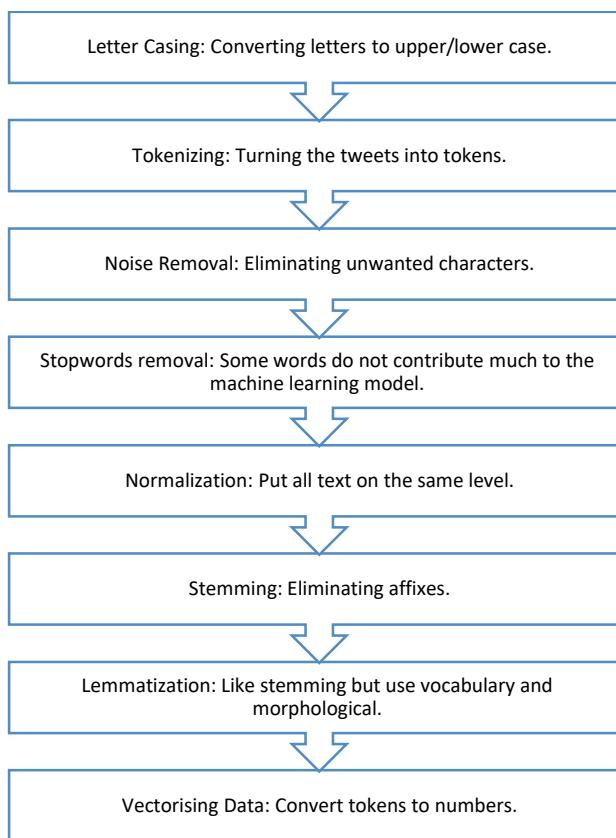


Figure 19 Data pre processing

The steps involved in pre-processing include converting letters to upper/lower case, tokenizing the text into individual units, removing unwanted characters and stopwords, normalizing the text, stemming to remove affixes, and lemmatization to reduce words to their base form. Finally, the pre-processed text is vectorized to convert the text data into numerical form that can be utilized by machine learning models.

2. Feature Extraction

The study employed three techniques to extract features from text data:

I. TF-IDF algorithm (term frequency-inverse document frequency): is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document and the inverse document frequency of the word across a set of documents. To implement the TF-IDF technique, the implementation of the TF-IDF technique utilized a class from the sklearn library.

II. word2vec: is a predictive model for computing a continuous radial representation of quantities in large data sets. The given models use two alternative models to get a high dimensional vector for each word:

- PCA: A technique focused on reducing the dimensions of words that directly impact how the original set of vectors transforms into a new set..
- t-SNE: A technique for nonlinear dimensionality reduction and data visualization. It combines words from a higher dimension with ones from a lower dimension. The Gensim library was used to construct word vectors using word2vec, with parameters like tokenized words and mincount set accordingly.

III. BERT: is a pre-trained language model for deep, bidirectional representations of unlabeled text by co-adapting on both the left and right context in all layers. BERT can be used in a variety of language tasks, with only a small layer added to the base model. BERT was used in two ways:

- Use the hugging face BERT model to fine-tune our sentiment analysis.
- Use the BERT model for fine-tuning and training on our dataset.

3.2.1.2.2 Impact on Current Study: Refinement of NLP Techniques for Digital Content Sentiment Analysis

Machine Learning Algorithms

Various machine learning can be used to accomplish the task of sentiment analysis.

The following machine learning algorithms are used:

The sentiment analysis task involved the utilization of a diverse set of machine learning and deep learning algorithms:

1. **K-Nearest Neighbors (KNN):** Identify the group to which a new data point (tweet) belongs based on training data. If a new tweet is close to a negative group, it is classified as negative; if close to positive or neutral, the prediction is made accordingly.
2. **Multinomial Naïve Bayes:** Determine the probability of a tweet being positive, negative, or neutral based on its contents or words.
3. **Decision Tree:** Classify a tweet based on its features, i.e., the words it contains.
4. **Random Forest:** Constructed from multiple decision trees to provide a more accurate and stable prediction. Operates as an ensemble, potentially offering improved results compared to a single decision tree.
5. **Support Vector Machine:** Utilize a non-linear Support Vector Classification model to categorize a tweet into positive, negative, or neutral classes.
6. **Voted Classifier:** Ensemble approach incorporating KNN, Multinomial Naïve Bayes, Random Forest, and Support Vector Machine to enhance sentiment analysis predictions.
7. **LSTM:** Leverage deep learning with LSTM, specifically effective for sequential data like text. LSTM's capacity to remember long-term dependencies in words contributes to its success in handling text sequences.

8. **Transformer Network:** Implementation of an encoder-only transformer model and the use of a pre-trained transformer model for sentiment analysis.
9. **K-Means Clustering:** Segregate groups with similar traits and assign them into clusters.

Model Training and Evaluation

- Training: Model training is an integral part of the whole process. It is very important to set the hyper-parameters of the models to the right ones to achieve good results.
- Evaluation: As we compare different machine learning models for sentiment analysis, various evaluation metrics will be employed to determine the model's performance.
 - Confusion Matrix.
 - 1. Precision.
 - 2. Recall.
 - 3. F1 Score.
 - 4. AUC-ROC Curve.

Model Testing

After training and evaluating the model, the best-performing model will be selected for testing with new tweets from Twitter. This process can be implemented using the Twitter API to create a user-friendly web server. Users can enter a keyword and date range, and the server will display the corresponding tweets along with their polarity. This provides organizations or individuals with valuable insights into what people are tweeting about their products or themselves.

Comparative Analysis:

A comparative analysis is provided for various sentiment analysis models and techniques applied to the digital media dataset. The objective is to assess and contrast the

performance of these models in terms of their ability to accurately classify sentiment, computational efficiency, and practical applicability in real-world scenarios.

Results

All models underwent training on 80% of our dataset, with the remaining 20% reserved for validation. Accuracy was employed as the performance metric during training, focusing on the validation accuracy of the models. The results are presented in the table (2) below.

Table 7 Validation Accuracy

Model	Validation Accuracy
K-Nearest Neighbors	55.36%
Multinomial Naïve Bayes	65.18%
Decision Tree	66.55%
Random Forest	70.82%
Support Vector Machine	65.98%
Voted Classifier	69.86%
LSTM	81.12%
BERT	94.56%

From the table above, it's evident that the BERT pre-trained model significantly outperformed even the LSTM model. Among the machine learning models, the random forest exhibited superior performance compared to others, including the voting classifier.

Evaluation:

Additional metrics, such as Confusion Matrix, Recall Score, Precision Score, and F1 Score, were employed to evaluate and compare the sentiment analysis models. The evaluation results for the aforementioned algorithms based on these criteria are presented in Figure (4) and Table (3).

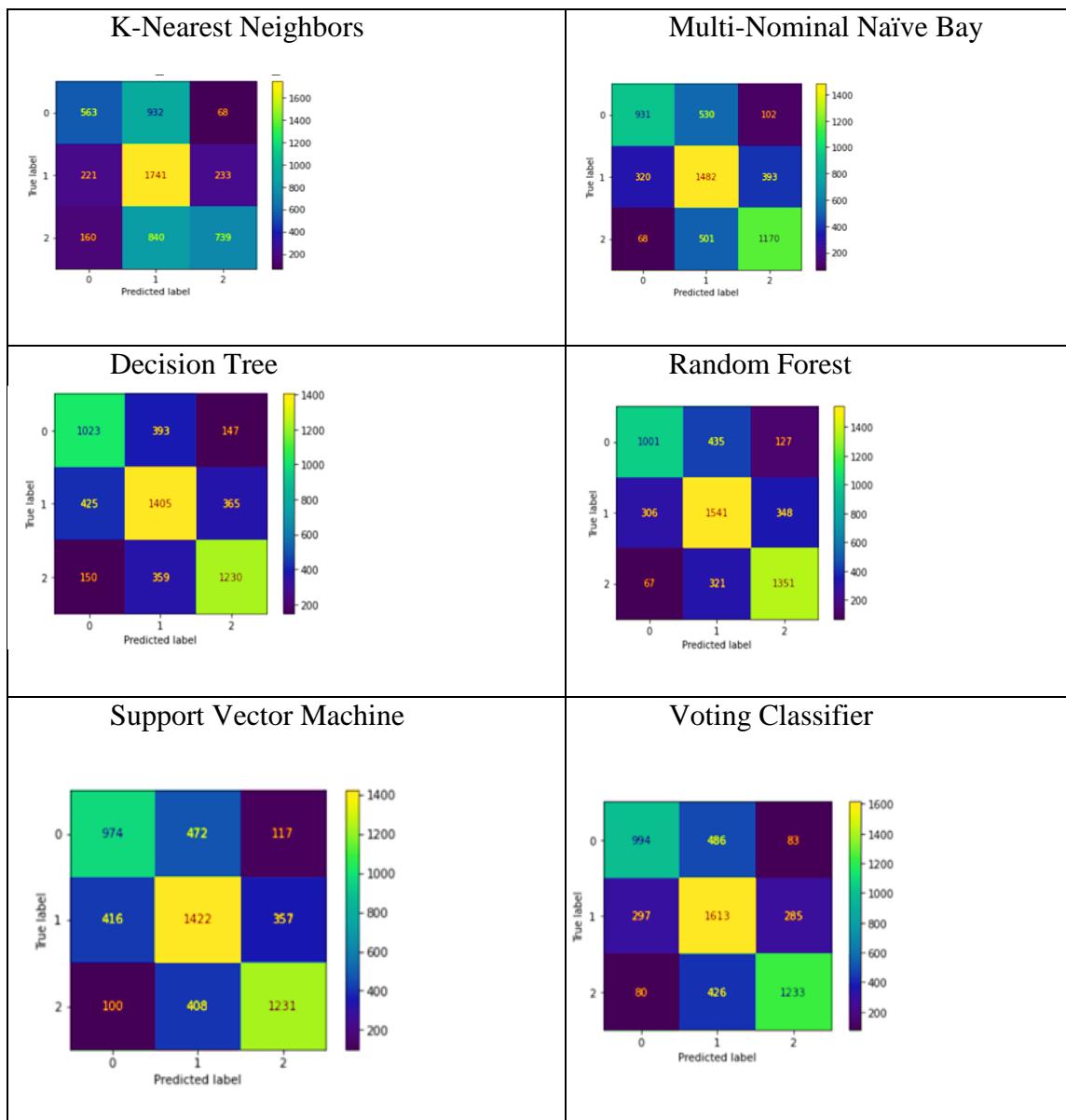


Figure 20 Confusion matrix

From the confusion matrix, it can be seen that most of the algorithm is best at classifying neutral tweets and they have a tendency to classify other tweets as neutral also.

Table 8 Recall, Precision, F1 Score of all algorithms

Model	Recall	Precision	F1 scores
K-Nearest Neighbors	0.5536	0.5923	0.5436
Multinomial Naïve Bayes	0.6518	0.6584	0.6321
Decision Tree	0.6655	0.6655	0.6509
Random Forest	0.7082	0.7091	0.6904
Support Vector Machine	0.6598	0.6609	0.6566
Voted Classifier	0.6986	0.7049	0.6921
LSTM	0.8112	0.8232	0.8012
BERT	0.9455	0.9551	0.9499

The table shows how well different machine learning algorithms performed on a task, based on three metrics: recall, precision, and F1 score. The LSTM and BERT models had the highest scores, indicating they were the most effective algorithms.

Proposed method-based deep learning algorithm (RNN-LSTM)

This method aims to enhance the preceding approach through an exploratory analysis of data to extract features, followed by the application of a deep learning algorithm to classify tweets into positive, negative, or neutral categories. Figure (5) illustrates the proposed framework for sentiment analysis. The improved method will be applied to the existing data for comparative evaluation with the previous approach, assessing the effectiveness of the proposed method.

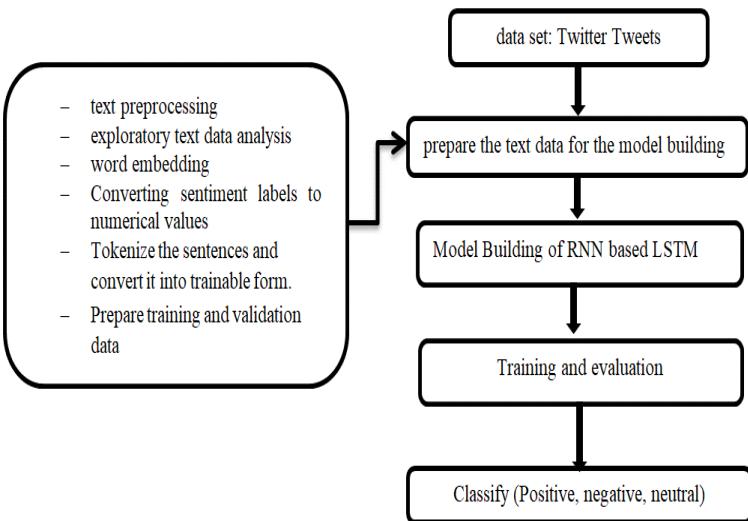


Figure 21 Framework for proposed Model

Algorithms

Step 1: prepare the text data for the model building

1. Pre-processing of text. It represents the initial phase of NLP projects. Some of the pre-processing to use a text cleaning procedure to clean all the content: Stop words, URLs, and punctuation such as! \$() *% @, lowercase, stemming, tokenization, and lemmatization have been removed.
2. Exploratory text data analysis is a straightforward yet incredibly informative method. In order to better understand the basic traits of the text data, it comprises (word frequency analysis, sentence length analysis, average word length analysis, distribution of the number of words, etc.). For instance:
 - The distribution of the number of words for each sentiment. so that we will use these features in the model training process.

- Distribution of the number of letters for each sentiment.
 - The count of the most frequent words in the entire text is essential for reinforcing the analysis in the feature extraction phase.
3. Apply word embedding to improve the model accuracy.

A word embedding is a type of learned representation of text in which words with the same meaning are represented similarly. It is regarded as one of the key developments in deep learning of document and word encoding for challenging natural language processing problems.

Word embedding is a method where individual words are represented as real-valued vectors in a specified vector space. Since each word is assigned to a distinct vector and the vector values are learned similarly to a neural network, the technique is frequently referred to as deep learning. Through the use of word embedding techniques, a corpus of literary works is used to learn a real-valued vector representation for a preset set-sized vocabulary. Other tasks, such as document categorization, include either unsupervised learning using document data or learning in conjunction with a neural network model. The word embedding method was developed using the Gensim library's keyed vectors library.

LSTMs employ a number of "gates" that regulate how data in a sequence enters, is stored in, and leaves the network. A typical LSTM has three gates: an output gate, an input gate, and a forget gate. Each of these gates is a separate neural network and may be thought of as a filter.

In order to avoid overfitting, the **Dropout** layer randomly sets input units to 0 with a frequency of rate at each step during training. Keep in mind that the Dropout layer only functions when the model's training is set to true, preventing any values from being dropped during inference.

A **dense** layer is densely connected to the layer above it is one in which every neuron in the layer is coupled to every other neuron in the layer above. The majority of artificial neural network networks employ this layer.

The values are unrolled starting with the last dimension when using the **flatten** operator.

Dropout layers: Since a dropout layer doesn't have any weights, it lacks parameters. A dropout layer only increases the likelihood that a neuron won't be tested by 1%. In a dropout layer, nothing more needs to be configured. We successively import layers, constants, dense, embedding, flatten, and initializers from keras in order to build the model.

Step 3: Model training

The number of samples that must be processed before the internal model parameters are changed is determined by the hyperparameter known as batch size. A for-loop is a type of batch that makes predictions while iterating through one or more samples. At the end of the batch, the predictions are compared to the expected output variables, and an error is then calculated. This issue is fixed using the updated algorithm, for instance by lowering the gradient of the error. The number of epochs hyperparameter controls how many times the learning algorithm will run through the entire training dataset.

For every sample in the training dataset, the internal model parameters changed once throughout an epoch. An era is made up of one or more batches. For instance, a single-batch epoch is described by the batch gradient descent learning process.

Step 4: Model Accuracy

Model accuracy is a measure of the proportion of correct predictions made by a model out of the total number of predictions produced. This metric is commonly used to assess a model's performance, although other metrics may also be considered. We found that this model can get 96% accuracy and this is better than our previous analysis.

Step 5: Plotting and Displaying Results

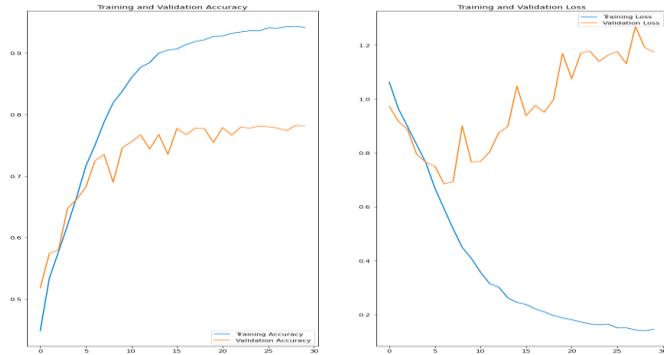


Figure 22 . Plot of Results of RNN Model

From Fig, During the sentiment analysis model's training phase, the loss and AUC metrics are shown for the training and validation sets. During the first 10 epochs, the loss is greatly reduced while the precision is noticeably improved.

Step 6: Based on their shared characteristics, various data subsets are segmented through clustering. Python offers a wide array of useful cluster analysis tools, and the choice of strategy depends on the specific task and the nature of the available data. Commonly utilized techniques in Python include Gaussian mixture models, spectral clustering, and K-means clustering. In this scenario, the K-means clustering method is employed. K-means clustering, a type of unsupervised machine learning, exclusively trains on inputs without generating outputs. It identifies distinct clusters of data points that are closest to each other. Once the data is partitioned into clusters, each point is assigned to the cluster whose mean is closest to that specific data point. We employ K-means clustering to create sentiment-based clusters from our data. For the implementation of sentiment analysis on the tweet data, a comprehensive pipeline has been developed. The model assigns different tweets to each of our clusters, encompassing three distinct sentiment labels.

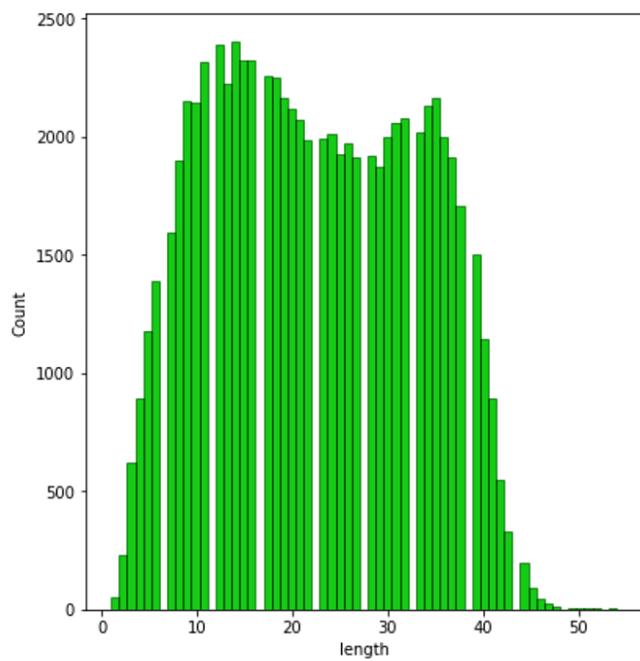


Figure 23 Positive sentiments tweets

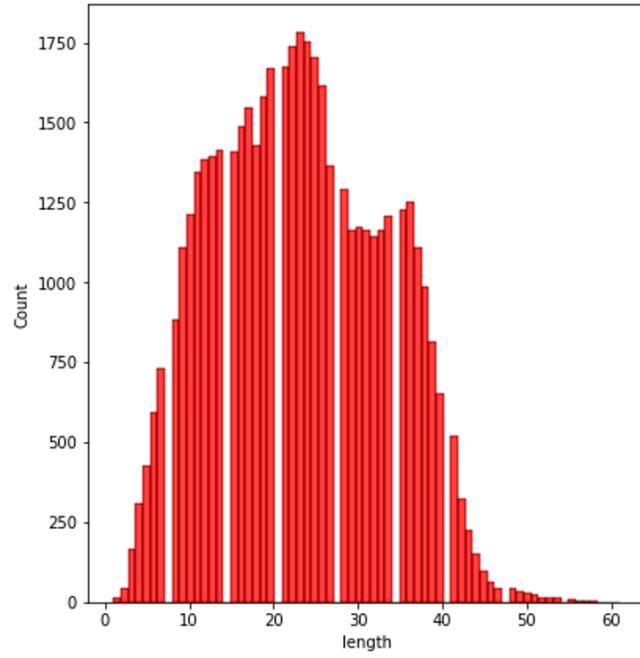


Figure 24. Negative sentiments tweets

So, the clustering algorithms assign a numerical value to each of the labels and assign a cluster based on its text context.

Summary of Results

The following are the main conclusions from the comparison of machine learning and deep learning algorithms for tweet sentiment analysis:

1. The BERT pre-trained model outperformed the LSTM model, achieving the greatest accuracy of 94.56%. This demonstrates that transfer learning for text classification problems can be quite successful when employing pre-trained language models.
2. With an accuracy rate of 70.82%, the Random Forest algorithm outperformed the other machine learning models. The method used in K Nearest Neighbours has the lowest accuracy, 55.36%.
3. The LSTM model received the greatest ratings for Recall, Precision, and F1, with values of 81.12%, 82.32%, and 80.12%, respectively, suggesting that it performed the best overall.
4. By achieving 96% accuracy, the suggested RNN-LSTM model with word embedding, dropout, and clustering proved the value of deep learning for sentiment analysis.
5. The model performance was evaluated thoroughly and rigorously utilizing confusion matrices, recall, precision, and F1 scores.

Utilizing a range of machine learning and deep learning techniques on Twitter data, our aim was to enhance decision-making and content quality in the dynamic digital media landscape. Key findings highlight BERT's exceptional 94.56% accuracy, showcasing transfer learning's effectiveness. Noteworthy results include the Random Forest algorithm (70.82% accuracy) and the LSTM model, excelling in Recall, Precision, and F1 scores. Deep learning, exemplified by the RNN-LSTM model, demonstrated exceptional potential with a 96% accuracy, establishing LSTM and BERT as tweet sentiment analysis frontrunners. Future

research should focus on refining models for real-world applications, exploring optimization, alternative architectures, and dataset expansion. Acknowledging study limitations points toward opportunities for a more nuanced approach within the digital media landscape.

In summary, this study lays the foundation for leveraging advanced sentiment analysis techniques, emphasizing the pivotal role of deep learning models, while recognizing the evolving nature of research in this domain.

3.2.2 Integration into Current Research Framework

In integrating methodologies from the prior research papers into the current study's framework, a focused effort has been made to harness the strengths and adapt the approaches to suit the specific requirements of sentiment analysis within digital media content. The methodologies outlined in the previous works, notably the utilization of Naive Bayes, RNN algorithms for real-time sentiment analysis for chatgpt, and the application of NLP techniques for digital media content analysis, served as foundational pillars guiding the design of the current research. These methodologies have been seamlessly integrated, serving as the blueprint for refining the current framework.

By strategically adapting these methodologies to accommodate the distinctive characteristics and intricacies of digital media content, the research framework has been honed to facilitate a more nuanced and contextually relevant approach to sentiment analysis. This integration process has not only informed the selection of algorithms and techniques but also guided the strategic customization of methodologies to align with the unique objectives and intricacies of the present study, thereby enriching the robustness and relevance of the current research framework.

3.2.2.1 Adaptation of Algorithms and Frameworks

The methodologies outlined in the previously published papers, encompassing a wide array of algorithms such as K-Nearest Neighbors, Multinomial Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine, Voted Classifier, LSTM, and BERT, have been instrumental in informing the algorithmic selection and framework development for the current study on sentiment analysis within digital media content.

In "A Novel Approach to Predict the Real-Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE," the focus on Naive Bayes and RNN algorithms for real-time sentiment analysis served as an initial blueprint. The adaptability and agility of these algorithms in handling real-time data streams and evolving sentiment patterns during a significant event like the COVID-19 pandemic inspired the exploration and customization of similar algorithms for the current research.

Simultaneously, in "Sentiment Analysis Predictions in Digital Media Content Using NLP Techniques," the utilization of K-Nearest Neighbors, Multinomial Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine, Voted Classifier, LSTM, and BERT algorithms showcased the diversity and effectiveness of various machine learning and deep learning techniques in deciphering sentiment within specific digital content domains.

Drawing from the methodologies employed in both papers, a strategic amalgamation of algorithms has been curated for the current study. The adaptability and versatility of these algorithms have been harnessed to cater to the intricacies of sentiment analysis across diverse digital media formats and platforms. Each algorithm's strengths have been leveraged and fine-tuned to handle the multifaceted nature of textual, visual, and multimedia content prevalent in digital media spaces. This integration not only broadens the spectrum of analysis

but also enhances the research framework's capacity to capture nuanced sentiments expressed within the complex landscape of digital media content.

3.2.2.2 Tailoring Methodologies to Suit Digital Media Context

The methodologies derived from prior research endeavors have been meticulously tailored and fine-tuned to suit the distinctive intricacies inherent in analyzing sentiment within the realm of digital media content. In previous studies, the application of sentiment analysis algorithms, such as Naive Bayes, RNN, K-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, Voted Classifier, LSTM, and BERT, primarily targeted specific contexts, necessitating a reconfiguration and recontextualization to align with the complexities of digital media platforms and content types.

The adaptability of these methodologies has been instrumental in addressing the multifaceted nature of digital media content, including textual, visual, and multimedia elements. Techniques that proved effective in analyzing sentiment within textual content in prior research were extended and modified to accommodate the diverse modes of expression prevalent in visual and multimedia formats across various digital platforms.

Moreover, considerations for context, tone, linguistic nuances, and evolving trends within digital media content were paramount in tailoring these methodologies. As digital media content often involves user-generated data and ever-evolving language patterns, the methodologies underwent enhancements to capture the dynamism and idiosyncrasies inherent in this landscape.

By fine-tuning these methodologies to the specific demands of digital media contexts, the research framework gained the capacity to navigate the intricate interplay of various

content types, platforms, and user-generated expressions. This tailored approach ensures a more nuanced and contextually relevant analysis of sentiments expressed within the multifarious landscape of digital media content, consequently enhancing the precision and applicability of the current study's methodologies.

3.2.2.3 Enhancements and Modifications for Specific Research Objectives

The methodologies inherited from previous research have undergone deliberate enhancements and modifications to align with the specific research objectives centered around the utilization of multiple digital media datasets procured from diverse resources. In prior studies, the algorithms and techniques, inclusive of Naive Bayes, RNN, K-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, Voted Classifier, LSTM, and BERT, were primarily applied within specific contexts or datasets.

The augmentation of these methodologies involved crucial adaptations to address the intricacies presented by the integration of diverse digital media datasets. Considerations for the heterogeneity of data sources, encompassing textual, visual, and multimedia content from various platforms, were paramount. Enhancements were implemented to ensure the seamless integration and analysis of these disparate datasets without compromising the integrity and accuracy of the sentiment analysis process.

Moreover, modifications were instituted to facilitate a cohesive approach to analyze sentiments across varied data sources. These modifications encompassed data preprocessing techniques, feature engineering strategies, and algorithmic adjustments tailored to accommodate the idiosyncrasies of each dataset while striving for a unified analysis framework.

By enhancing and modifying the inherited methodologies to suit the incorporation of multiple digital media datasets from various resources, the research framework now stands equipped to tackle the complexities inherent in amalgamating diverse data sources. This strategic approach enables a more comprehensive and inclusive sentiment analysis across a spectrum of digital media content, thereby fortifying the capacity to derive nuanced insights aligned with the specific research objectives.

3.2.3 Refinement and Customization of Methodologies

3.2.3.1 Modifications Based on Data Source Characteristics

The methodologies inherited from prior research underwent meticulous refinement and adaptation, considering the diverse characteristics inherent in the datasets sourced from blogs, Spotify, Twitter, and Facebook. Each data source presents unique challenges related to data structure, content format, user interactions, and context. To accommodate these intricacies, modifications were implemented in preprocessing techniques, feature extraction methods, and algorithmic configurations.

Tailored modifications were made to account for the unstructured nature of blog data, the audio-centric content from Spotify, the succinct and evolving nature of tweets on Twitter, and the multi-modal and diverse interactions present in Facebook content. These modifications aimed to harmonize the methodologies to effectively process and analyze data from each source while ensuring consistency and reliability in sentiment analysis.

3.2.3.2 Customization to Address Unique Aspects of Digital Media Content

The customization of methodologies was further extended to address the unique aspects prevalent in digital media content. This included considerations for multimedia content formats, such as audio from Spotify, textual content from blogs and tweets, and the amalgamation of textual, visual, and interactive elements in Facebook data.

Techniques were customized and optimized to handle the varying characteristics and modalities of digital media content. For instance, algorithms were fine-tuned to decode sentiments from textual content while simultaneously accounting for emotional cues conveyed through audio content on Spotify. Moreover, sentiment analysis frameworks were adapted to capture nuanced sentiments expressed through visual elements or interactive features within Facebook content.

3.2.3.3 Aligning Methodologies with Current Research Goals

These refined and customized methodologies were strategically aligned with the current research goals, which encompassed comprehensive sentiment analysis across diverse digital media platforms. By integrating and harmonizing methodologies specifically tailored to suit the characteristics of data sourced from blogs, Spotify, Twitter, and Facebook, the research framework was positioned to effectively achieve the overarching research objectives of holistic sentiment analysis within the multi-faceted landscape of digital media content.

3.4 Quantitative Techniques: Numerical Analysis through AI Tools

Within the framework of this research, quantitative analysis techniques have been inspired and refined by the methodologies employed in the published work, particularly leveraging Naive Bayes, RNN algorithms, and Natural Language Processing (NLP) techniques. The insights garnered from these published papers have guided the development and implementation of similar quantitative approaches in this study for sentiment analysis and prediction within digital media content.

3.5 Justification: Reasoning Behind Methodological Choices

The methodologies utilized in the published research papers have been critically examined and justified in alignment with the current research objectives. The adaptation of certain techniques from these papers has been rationalized based on their effectiveness in real-time sentiment analysis and their applicability to the specific context of digital media content in this study.

3.5.1 Methodological Decision-Making: Rationale for Approaches

Methodological decision-making involves elucidating the rationale behind the chosen approaches and strategies for conducting the research. It aims to provide a clear understanding of why specific methodologies, tools, and techniques were selected over others. This section of the methodology chapter outlines the thought process and reasoning behind methodological choices, including the selection of algorithms, data collection methods, and analysis techniques for sentiment analysis within digital media content.

Key elements to discuss under this section include:

- **Justification for Method Selection:** Explain why particular algorithms (such as Naive Bayes, RNN, etc.) and tools were chosen for sentiment analysis. Discuss their strengths, relevance to the research objectives, and applicability to the diverse nature of digital media content.
- **Rationale for Data Sources:** Describe the reasoning behind selecting blogs, Spotify, Twitter, and Facebook as data sources. Highlight the diversity and richness of these platforms and how they contribute to a comprehensive understanding of sentiment across various media types.
- **Considerations for Multi-source Analysis:** Justify the decision to employ multiple sources for a holistic approach to sentiment analysis. Explain how integrating various platforms enriches the analysis and captures diverse perspectives and expressions.

- **Framework Flexibility:** Discuss the flexibility of the chosen methodologies to adapt to the dynamic nature of digital media content. Explain how these approaches accommodate different data formats, languages, and evolving trends in online communication.

3.5.2 Ethical Considerations: Protecting Participants' Rights

Ethical considerations are paramount when conducting research involving data from online platforms and user-generated content. This section focuses on addressing the ethical principles and safeguards implemented to ensure the protection of participants' rights and confidentiality.

Key points to address within this section include:

- **Informed Consent:** Describe how participant consent was obtained, if applicable. Explain measures taken to inform and obtain consent from individuals whose data is utilized in the study, ensuring transparency and voluntariness.
- **Anonymization and Privacy:** Discuss steps taken to anonymize and protect the privacy of individuals contributing data to maintain confidentiality and prevent any potential harm or identification.
- **Compliance with Regulations:** Highlight adherence to legal and ethical guidelines, such as GDPR, HIPAA, or other relevant regulations, ensuring that the research complies with ethical standards and safeguards participants' rights.
- **Data Security and Storage:** Explain protocols for secure data handling, storage, and disposal to prevent unauthorized access and maintain data integrity throughout the research process.

3.6 Novel Methodological Framework for Comprehensive Exploration

This study aims to propose a novel approach for predicting sentiments from digital media user reviews by integrating text sentiment analysis and converting video and audio data into textual format for comprehensive sentiment understanding.

The chosen research approach encompasses a mixed-method methodology that amalgamates quantitative and qualitative paradigms. This hybrid approach is deemed suitable as it allows a holistic examination of the opinion mining implementation on digital media. It facilitates numerical analysis while embracing the richness of qualitative insights from user perspectives and content nuances.

Understanding the Hybrid Paradigm

The selection of a mixed-method methodology, fusing both quantitative and qualitative paradigms, epitomizes a conscious endeavor to harness the strengths of divergent research methodologies. This hybrid approach converges numerical analysis and qualitative insights, underpinning a holistic examination of opinion mining implementation within the dynamic landscape of digital media (Rigaki, 2023).

Embracing Quantitative Analysis

Quantitative methodologies traditionally thrive on empirical observations, statistical analysis, and numerical data interpretation. Within the context of implementing opinion mining on digital media, quantitative analyses afford the ability to quantify sentiments, preferences, and trends across vast datasets. Statistical models, machine learning algorithms, and sentiment analysis metrics serve as the bedrock for objectively measuring sentiments and identifying patterns within digital content.

Leveraging Qualitative Insights

In stark contrast, qualitative paradigms delve into the intricate nuances, subjective perceptions, and contextual intricacies embedded within data. When applied to the realm of

digital media content, qualitative methodologies entail understanding user perspectives, decoding semantic nuances, and interpreting the underlying reasons behind expressed sentiments. Interviews, surveys, and content analysis aid in unraveling the richness of user experiences, shedding light on the multifaceted nature of opinions within digital content.

Synergy between Quantitative and Qualitative Approaches

The fusion of these disparate methodologies within a mixed-method framework for opinion mining implementation on digital media brings forth a synergistic interplay. Quantitative analysis, through sentiment analysis algorithms and statistical models, provides a foundational understanding of sentiment distribution, polarity, and trends across digital content. Simultaneously, qualitative insights gleaned from user interviews, content analysis, and surveys enrich the interpretation of sentiments, offering context-specific elucidation of opinions embedded within digital media.

Holistic Examination of Opinion Mining on Digital Media

The amalgamation of quantitative and qualitative paradigms allows for a comprehensive exploration of opinion mining implementation on digital media. While quantitative analyses yield empirical evidence and trends, qualitative insights contribute depth and context, elucidating the "why" behind sentiments expressed within digital content. This comprehensive approach not only captures the breadth of opinions but also delves into the underlying rationales, facilitating a nuanced understanding of user sentiments and content nuances.

Advantages and Suitability of the Hybrid Approach

The suitability of this hybrid approach lies in its capability to mitigate the limitations inherent in singular methodologies. While quantitative analyses might overlook contextual intricacies, qualitative insights bridge this gap by providing a deeper understanding of user behaviors and motivations. Conversely, qualitative analyses might lack generalizability,

which quantitative analyses complement by offering broader trends and statistical significance.

The Synergistic Exploration

the adoption of a mixed-method methodology integrates the strengths of quantitative and qualitative paradigms, fostering a synergistic exploration of opinion mining implementation on digital media. This hybrid approach not only augments the breadth and depth of insights but also ensures a more robust and comprehensive understanding of user sentiments, preferences, and content nuances within the dynamic landscape of digital media platforms.

This excerpt outlines the scoping study focused on assessing the level of AI-based sentiment analysis in contemporary competitive research. Let's delve deeper into the explanation of each research question (RQ):

Research Question 1: How is the current state of research?

This question seeks to comprehensively assess the current landscape of research in AI-based sentiment analysis concerning competitive research. It involves:

- **Literature Review Scope:** Conducting a thorough review of existing literature from databases like Scopus to identify and synthesize the current state of research in this domain.
- **Trends and Patterns:** Analyzing trends, methodologies, and key findings from the collected literature to understand the prevailing approaches, challenges, and advancements.

Research Question 2: What is the development of the major AI-based sentiment analysis approaches?

This question aims to elucidate the evolution and progression of major AI-based sentiment analysis methods within the competitive research sphere. It includes:

- **Methodological Evolution:** Tracing the development and evolution of sentiment analysis techniques leveraging AI within competitive research settings.
- **Technological Advancements:** Exploring the adoption of AI algorithms, neural networks, natural language processing (NLP) techniques, and machine learning models in sentiment analysis.

Research Question 3: What are the difficulties and prospects of AI-based sentiment analysis in competitive research?

This question focuses on identifying challenges and potential opportunities for AI-based sentiment analysis within the competitive research landscape. It involves:

- **Identification of Challenges:** Investigating the hurdles, limitations, and barriers encountered in applying AI-based sentiment analysis techniques in competitive research contexts.
- **Exploration of Prospects:** Identifying potential growth areas, novel applications, and future directions for leveraging AI-based sentiment analysis effectively in competitive research, considering emerging technologies and methodologies.

Why is Sentiment Analysis Important, Existing Methods, and Improvement Strategies?

Importance:

Sentiment analysis provides invaluable insights into public opinions, aiding businesses, policymakers, and researchers in decision-making, marketing strategies, and understanding social trends.

Existing Methods:

Current methods encompass rule-based systems, machine learning algorithms (like Naive Bayes, SVM, and neural networks), and lexicon-based approaches.

Improvement Strategies:

To enhance existing methods, consider:

- **Deep Learning Models:** Employing advanced neural network architectures (LSTMs, Transformers) for better context understanding.
- **Multimodal Fusion:** Integrating information from text, images, and videos for a more comprehensive analysis.
- **Contextual Understanding:** Incorporating contextual information to discern nuances better.

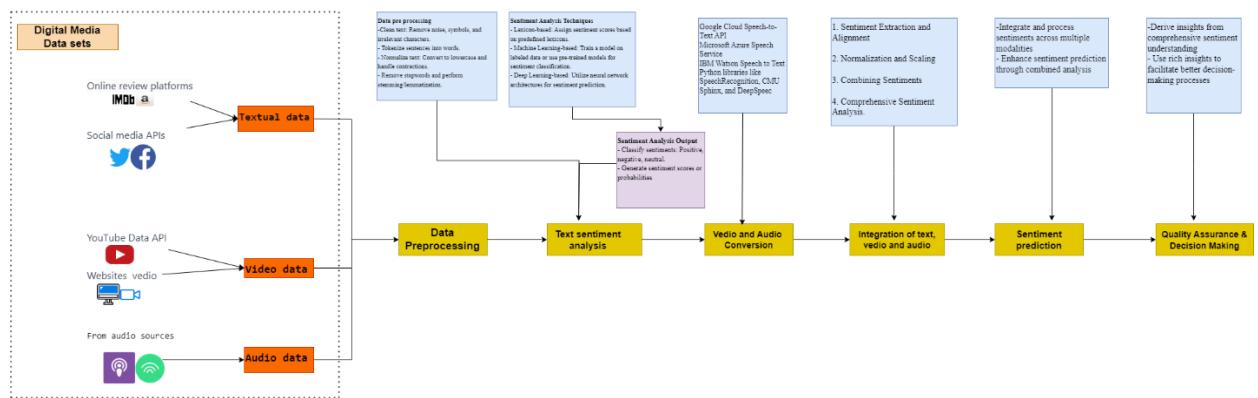


Figure 25 our novel framework for Multimodal Sentiment Analysis:

3.6.1 Data Collections

1. **Textual Data:** Collect user reviews, comments, and textual content from various platforms such as social media, review websites, and forums.
2. **Video Data:** Extract audio content from videos using speech-to-text conversion techniques to convert spoken words into textual data.
3. **Audio Data:** Utilize speech recognition and transcription tools to convert audio content (podcasts, interviews, etc.) into textual format.

How to Collect the Data?

- **API Access:** Utilize APIs provided by social media platforms to access and gather user-generated content.

- **Web Scraping:** Employ web scraping techniques to collect data from review websites.
- **Dataset Download:** Download datasets from public repositories or request access from data providers.

3.6.3 Approach:

1. Text Sentiment Analysis:

- Utilize Natural Language Processing (NLP) techniques such as sentiment lexicons, machine learning models, or deep learning models to perform sentiment analysis on textual data.
- Classify sentiments (positive, negative, neutral) from user reviews, comments, and textual content.

In the process of text sentiment analysis, data preprocessing is crucial for preparing textual data for analysis.

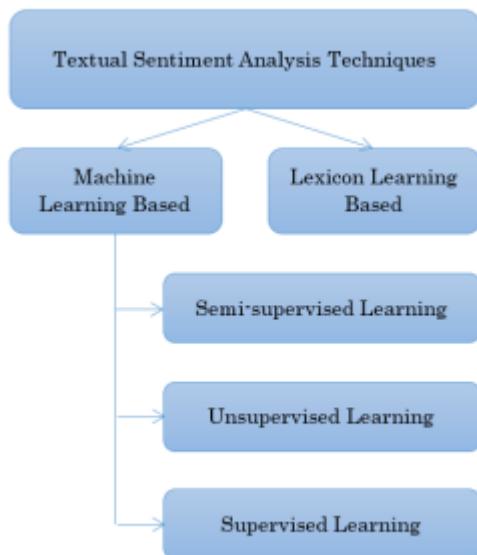


Figure 26 Textual sentiment analysis techniques

This involves cleaning the text by removing irrelevant characters and symbols, tokenization to break down sentences into individual words, normalization for consistency, stopword removal to eliminate common words, and stemming or lemmatization to reduce words to their base form. Sentiment analysis techniques encompass lexicon-based approaches using sentiment dictionaries, machine learning-based methods employing models like Support Vector Machines or Naive Bayes, and deep learning-based approaches utilizing neural network architectures such as RNNs, LSTMs, or Transformer models like BERT and GPT. The sentiment analysis output typically involves classifying sentiments into categories like positive, negative, or neutral, often with a graded scale to capture subtleties.

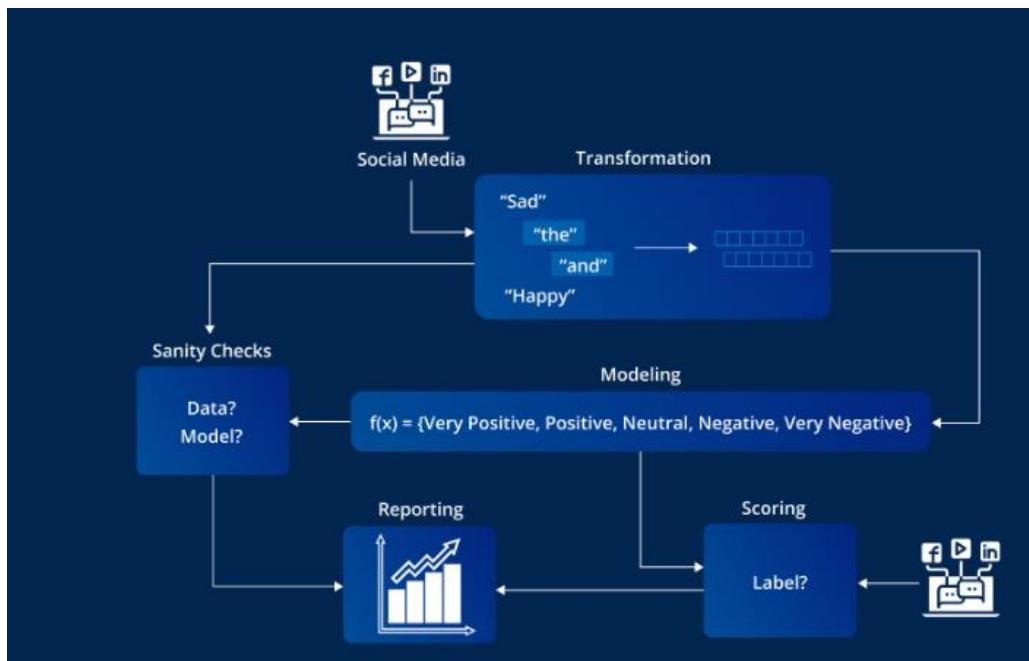


Figure 27 framework to collect text from social medias

Additionally, sentiment scores or probabilities are generated to quantify the likelihood of each sentiment class, providing a nuanced understanding of the text's emotional tone. Implementing these steps involves choosing appropriate libraries or tools (such as NLTK, spaCy, scikit-learn, TensorFlow, or PyTorch) in a programming language like Python to perform data preprocessing and apply sentiment analysis techniques on your textual data.

2. Conversion of Video and Audio Data into Text:

In the process of converting video and audio data to text, the first step involves speech-to-text conversion, which can be achieved through speech recognition software or APIs.



Figure 28video to audio then text

This technology utilizes Automatic Speech Recognition (ASR) techniques to transcribe spoken words from audio in videos into textual transcripts. ASR algorithms are essential for processing audio signals and recognizing speech patterns, ultimately converting the audio data into written text. Alternatively, deep learning models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer models, play a crucial role in this conversion process. These neural network-based models are trained on speech data, enabling them to accurately transcribe spoken language in video content, providing a valuable means of converting audio information into a readable and searchable text format.

- Employ speech-to-text algorithms or pre-trained models for converting audio data into transcribed text.
- Extract spoken content from videos and podcasts to create textual representations of the audio.

For implementing these steps, various tools and APIs are available that provide speech recognition capabilities:

- Google Cloud Speech-to-Text API
- Microsoft Azure Speech Service
- IBM Watson Speech to Text
- Python libraries like SpeechRecognition, CMU Sphinx, and DeepSpeech

These tools and libraries facilitate converting audio content from videos into textual format, enabling further analysis, including sentiment analysis, based on the transcribed text data.

3.6.4 Data Preprocessing and Analysis:

1. Textual Data Processing:

- Preprocess textual data by cleaning, tokenization, and normalization to prepare it for sentiment analysis.
- Apply sentiment analysis techniques to identify sentiments in text (positive, negative, neutral).

2. Audio-to-Text Processing:

- Convert transcribed text from audio data into a format suitable for sentiment analysis.
- Apply the same sentiment analysis techniques used for textual data to the converted text from audio.

3.6.5 Integration and Sentiment Prediction:

1. Multimodal Integration:

- Integrate sentiments derived from text with sentiments obtained from the converted audio data.

- Combine and analyze sentiments across different modalities to get a holistic view of user sentiments.

Insights and Findings:

1. **Comprehensive Sentiment Understanding:** Obtain a more comprehensive understanding of user sentiments by analyzing sentiments from both textual and converted audio data.
2. **Correlation of Modalities:** Explore correlations and discrepancies between textual sentiments and sentiments derived from audio data.

The integration of text, video, and audio sentiments is a multifaceted process involving sentiment extraction, alignment, normalization, scaling, and comprehensive analysis. Sentiments derived from textual reviews and comments are extracted alongside sentiments obtained from converted audio data, where transcribed speech is analyzed. Aligning sentiments based on time stamps or contextual cues facilitates the correlation of sentiments from text and audio sources related to the same content. The normalization and scaling of sentiment scores or labels address variations in scales or scoring mechanisms, ensuring consistency and facilitating a unified analysis. The combined sentiments from different modalities are then merged, with the option to assign weights based on confidence levels or relevance, providing flexibility in emphasizing one modality over the other. A comprehensive sentiment analysis is conducted across different modalities, offering a holistic view of sentiments associated with the content. This integrated approach allows for the identification of correlations, contradictions, or complementary insights, contributing to a nuanced understanding of sentiments in a cohesive and unified manner.

Implementing this process would involve developing algorithms or scripts to extract sentiments from text and converted audio data, aligning sentiments, normalizing and scaling sentiment values, merging sentiments, and performing comprehensive sentiment analysis

across different modalities. Python or other programming languages can be used along with appropriate libraries or tools for data manipulation, alignment, and analysis.

3.6.6 Sentiment prediction

The sentiment prediction process begins with the integration and aggregation of sentiment data from diverse sources, including text analysis, video, and audio inputs, creating a cohesive dataset. This integrated data undergoes preprocessing, incorporating feature engineering and normalization to prepare for predictive modeling. Relevant features capturing sentiment nuances across modalities are extracted, and dimensionality reduction techniques are applied for efficient feature selection. Choosing an appropriate machine learning or deep learning model, such as Support Vector Machines, Random Forest, or Neural Networks, the model is trained on the integrated sentiment dataset, considering cross-modality correlations. Evaluation and validation of the model's performance follow, employing techniques like cross-validation and assessing metrics such as accuracy, precision, recall, and F1-score. The trained model is then utilized for sentiment prediction on new or unseen data, benefiting from the combined analysis of text, video, and audio sentiments to enhance accuracy and provide comprehensive insights across multiple modalities.

Pseudocode

```
# Step 1: Integrated Sentiment Data Preparation  
combined_data = combine_sentiments(text_data, video_data, audio_data)  
  
if combined_data is not empty:  
    preprocessed_data = preprocess_data(combined_data)  
  
else:  
    print("Error: Combined data is empty.")  
    exit()
```

```

# Step 2: Feature Extraction and Selection

if preprocessed_data is not empty:

    relevant_features = extract_features(preprocessed_data)

    if relevant_features is not empty:

        selected_features = select_features(relevant_features)

    else:

        print("Error: Relevant features are empty.")

        exit()

else:

    print("Error: Preprocessed data is empty.")

    exit()


# Step 3: Model Building and Training

selected_model = choose_model()

if selected_model is not None:

    trained_model = train_model(selected_model, preprocessed_data)

else:

    print("Error: Selected model is not valid.")

    exit()


# Step 4: Model Evaluation and Validation

if trained_model is not None:

    evaluation_results = evaluate_model(trained_model, preprocessed_data)

    validation_metrics = validate_model(trained_model, validation_data)

else:

```

```

print("Error: Model is not trained.")

exit()

# Step 5: Enhanced Sentiment Prediction

new_data = get_new_data()

if trained_model is not None and new_data is not empty:

    predicted_sentiments = predict_sentiments(trained_model, new_data)

    enhanced_prediction = combine_analysis(predicted_sentiments, text_data,
video_data, audio_data)

else:

    print("Error: Model or new data is not valid.")

    exit()

# End

print("Sentiment prediction process completed successfully.")

Implementing this process involves leveraging machine learning or deep learning frameworks in programming languages like Python. Utilize libraries such as scikit-learn, TensorFlow, or PyTorch for model building, feature extraction, training, evaluation, and prediction tasks. Ensuring the integration of sentiments across multiple modalities during feature extraction and model training will be crucial for enhanced sentiment prediction based on combined analysis.

```

3.6.7 Benefits:

Enhanced Sentiment Prediction: Improve sentiment prediction accuracy by integrating information from multiple modalities.

1. **Richer Insights:** Gain deeper insights into user sentiments by considering both text and audio-based reviews.
2. **Improved Decision Making:** Make more informed decisions based on a more complete understanding of user sentiments from diverse media sources.

This proposed approach of multimodal sentiment analysis, combining text sentiment analysis with the conversion of video and audio data into textual format, provides a more comprehensive understanding of user sentiments from digital media reviews. This holistic approach enhances sentiment prediction and enables better decision-making processes based on rich insights gathered from multiple modalities.

Conclusions

The conclusion section summarizes the comprehensive methodological framework designed for exploring sentiment within digital media content. It encapsulates the key aspects of the methodology, including the rationale behind method selection, ethical considerations, and the overall framework's suitability for achieving the research objectives.

In this section, reiterate the importance of the chosen methodologies and ethical considerations in conducting a robust and comprehensive analysis. Emphasize how the outlined framework enables a nuanced exploration of sentiments across diverse digital media platforms, contributing to the credibility and reliability of the research outcomes. Finally, provide a bridge to the subsequent chapters by highlighting how this methodological framework aligns with the research's broader goals and contributes to generating valuable insights in the field of sentiment analysis within digital media content.

Chapter IV.

Implementation

This chapter delineates the practical execution of the outlined methodology and conceptual framework in analyzing public sentiment regarding ChatGPT across diverse digital media platforms. The implementation phase signifies the pivotal transformation of theoretical constructs into actionable steps, integrating innovative opinion mining approaches driven by Artificial Intelligence (AI) to elucidate the nuanced perceptions of users.

4.1 Introduction

This chapter embodies the synthesis of methodologies acquired from sentiment analysis, machine learning, and data processing domains, specifically tailored to discern, categorize, and comprehend the multifaceted opinions expressed through tweets, comments, reviews, and posts across platforms such as YouTube, Twitter, Facebook, IMDb, Spotify, Amazon, and website videos.

4.2 Case Study: ChatGPT Tweets Sentiment Analysis

The focal point of this implementation involves an in-depth exploration of ChatGPT's reception and impact within the digital sphere, leveraging a case study centered on sentiment analysis of user-generated content across multiple digital platforms (Kumari, 2023). ChatGPT, an advanced language model developed by OpenAI, has garnered substantial attention and engagement across various online channels due to its versatile capabilities and widespread adoption (Su, 2022).

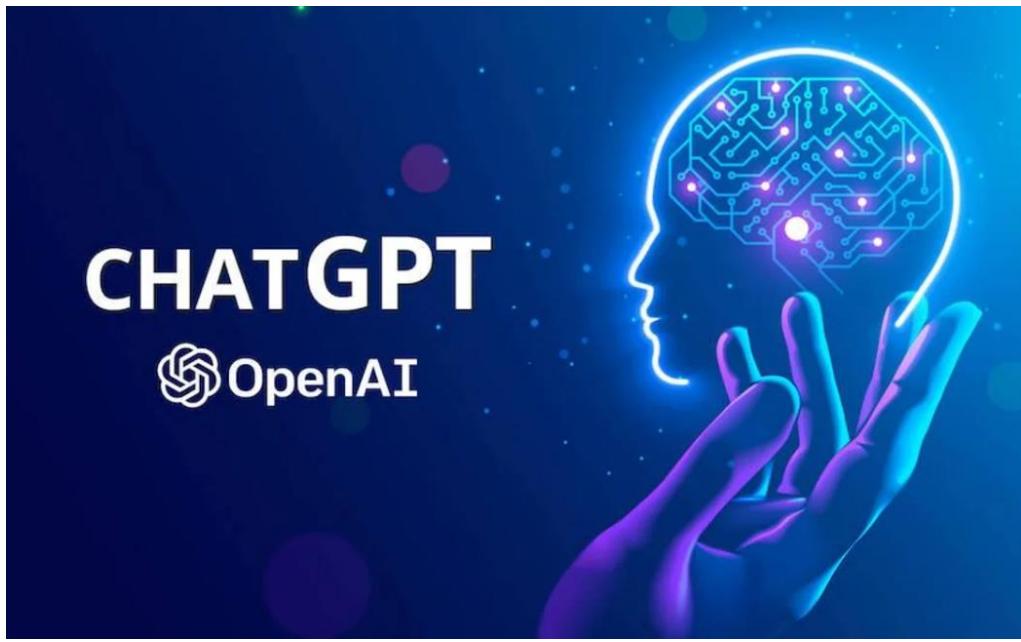


Figure 29 Chatgpt AI

The case study intricately scrutinizes the sentiments encapsulated within user interactions, opinions, and experiences pertaining to ChatGPT. By analyzing a diverse dataset collected from Twitter, IMDb, Amazon, Spotify, Facebook, and website videos, this study aims to unravel the overarching sentiment trends, prevalent topics of discussion, and underlying patterns within the user-generated content surrounding ChatGPT.

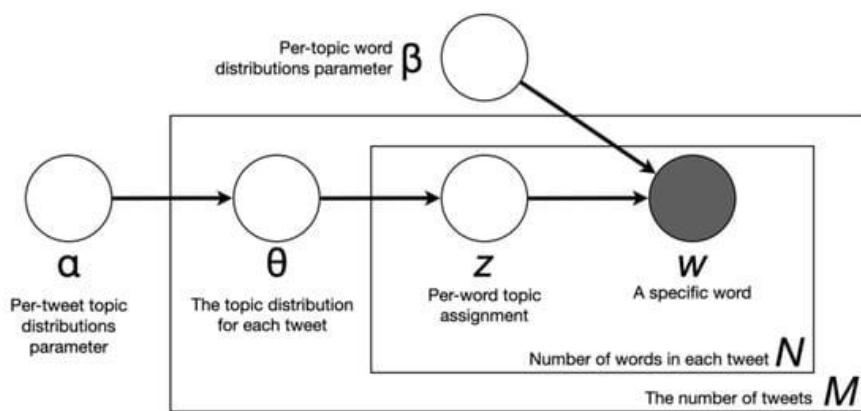


Figure 30 General framework for case study

Through the implementation of novel opinion mining approaches rooted in Artificial Intelligence, this case study endeavors to provide actionable insights that contribute to quality assurance strategies and informed decision-making processes. The utilization of sophisticated AI-driven sentiment analysis algorithms seeks to extract, categorize, and interpret sentiments expressed by users, thereby empowering a comprehensive understanding of the public's perceptions regarding ChatGPT's functionality, updates, performance, and user experience (Korkmaz, 2023).

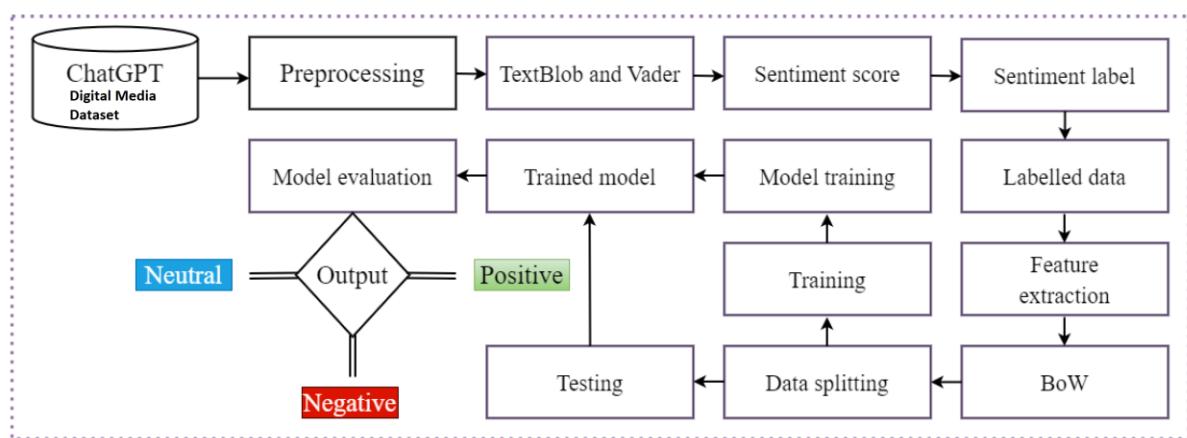


Figure 31 Novel method of case study

This chapter navigates the procedural steps undertaken in data collection, preprocessing, sentiment analysis, trend identification, and topic modeling, presenting a detailed account of the implementation process (Wang Z. Q., 2023). The amalgamation of empirical findings and analytical frameworks derived from this case study serves as a cornerstone for extrapolating actionable insights and recommendations in the subsequent chapters, contributing substantively to the field of AI-driven sentiment analysis in digital media contexts.

4.3 Dataset

Implementing a novel opinion mining approach using Artificial Intelligence (AI) on digital media content can significantly aid in quality assurance and decision-making processes.

Like Count	Content	User
64094	Writing erotic prompts for Chat GPT hoping to be the first person to make AI cum. It's like the Turing Test	MoistCr1TiKaL
63835	Best AI Tools You Need To Know #chatgpt #chatgpt3 #ArtificialIntelligence #ai https://t.co/0jfr8cOMoo	johnvianny
44940	I am pretty sure I am reading my first CHAT GPT student essay and like, teachers, don't panic. You'll know it when you see it.	rgay
42125	ultra-modern generative ai □: \n\n• AI2\n• AI21\n• mdm\n• gpt-J\n• gpt-3\n• x-clip\n• bloom\n• cohere\n• gopher\n• dalle 2\n• craiyon\n• tabnine\n• jukebox\n• chatGPT ***\n• anthropic\n• codegeex\n• nvidia get3d\n• dreamfusion\n• stable diffusion\n• meta make-a-video https://t.co/ON5eIGvnEQ	aaronsiim
38278	First #ChatGPT answer which made me pause. https://t.co/nbc9uRoWIA	kevinschawinski

Figure 32 Most liked words on digital media about ChatGPT from January to March 2023.

For your thesis, data collection from diverse social media platforms like YouTube, Twitter, Facebook, Spotify, IMDb, and others related to ChatGPT is crucial to achieve a comprehensive understanding of public sentiment. Utilizing APIs and web scraping techniques will allow you to extract a variety of user-generated content, including tweets, comments, reviews, and posts. These platforms host a spectrum of opinions, discussions, and user experiences concerning ChatGPT, enabling a multi-faceted analysis. To ensure the dataset's balance, it's imperative to gather a representative sample size from each platform, considering the differing user demographics, engagement styles, and content formats unique to each. This balanced dataset will serve as the foundation for your sentiment analysis,

facilitating a holistic examination of public perceptions, trends, and topics surrounding ChatGPT across multiple digital mediums. Additionally, the inclusion of a diverse dataset will enhance the robustness and validity of your findings, allowing for a more accurate portrayal of the sentiment landscape regarding ChatGPT on various social media platforms.

4.3.1 Overview of Updated Data Collection Methodologies

In this section, the updated approaches and methodologies employed for data collection during this phase of analysis are elucidated. The objective of this section is to highlight any refinements or augmentations made to the data collection process, focusing on ensuring a comprehensive and diverse dataset for sentiment analysis without the utilization of an AI detector (Parvin, 2023)

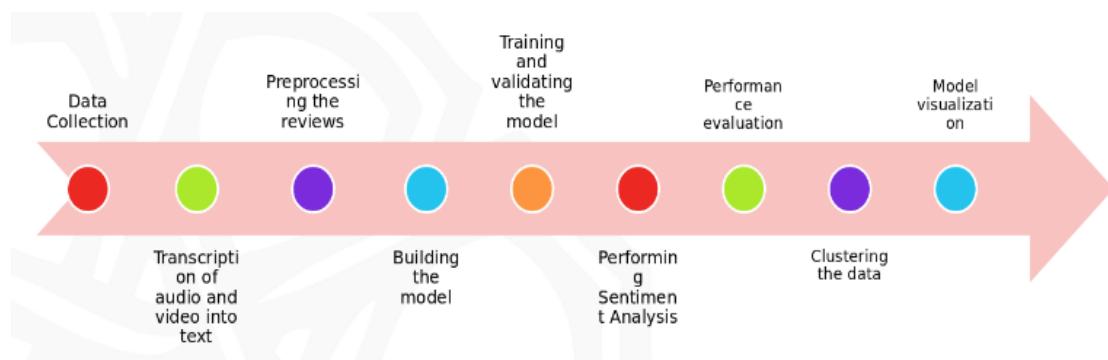


Figure 33 Data preprocessing and analysis

The data collection methodology remains consistent with the prior stages, encompassing the utilization of APIs, web scraping techniques, and manual curation to retrieve user-generated content across various digital media platforms, including Twitter, IMDb, Amazon, Spotify, Facebook, and website videos. However, unlike previous iterations where AI detectors were applied, the current phase excludes the use of such tools to gather content for this analysis.

The primary emphasis is on leveraging platform-specific APIs and custom scripts to collect public posts, comments, tweets, and reviews explicitly related to ChatGPT without relying on AI-based detection tools. This decision is made to ensure an unbiased approach to data collection, allowing for a diverse representation of user opinions and minimizing potential biases that could arise from AI-detected content.

Furthermore, human oversight and manual verification processes have been intensified to filter out irrelevant or unrelated content. A meticulous review of collected data has been conducted to ensure the dataset's relevance, accuracy, and adherence to the predefined criteria established for ChatGPT-related content (Mujahid, 2023).

The absence of AI detectors in this phase aims to provide a more organic and unfiltered representation of user sentiments and opinions across multiple platforms. This methodology adjustment strives to maintain the authenticity and integrity of the collected dataset while emphasizing human discretion and qualitative evaluation in the curation process (Küçük, 2019).

To leverage this approach effectively, diverse datasets from various digital media sources can be utilized. Here's how different datasets from platforms such as Twitter, Facebook, IMDb, YouTube, Amazon, and Spotify can be employed as follows:

Table 9 shows tweets, posts, or comments from Twitter, IMDb, Amazon, Spotify, Facebook, and a website video platform.

Tweet ID	Platform	User ID	Date & Time	Post Text	Sentiment	Topic/Category
1	Twitter	@user1	2023-01-05	"Loving the new ChatGPT update! 😊"	Positive	Features/Updates

2	YouTube	user123	2023-01-07	"This ChatGPT version seems inefficient."	Negative	Performance
3	Facebook	456789	2023-01-10	"ChatGPT's accuracy is remarkable."	Positive	Accuracy
4	Twitter	@user2	2023-01-12	"Struggling to understand ChatGPT's responses."	Negative	Comprehension
5	IMDb	user456	2023-01-15	"ChatGPT enhances user experience on the platform!"	Positive	User Experience
6	Amazon	customer A	2023-01-18	"ChatGPT-based recommendations are spot on."	Positive	Recommendations
7	Spotify	userXYZ	2023-01-20	"ChatGPT's music suggestions are fantastic!"	Positive	Music Recommendations
8	Twitter	@user3	2023-01-22	"ChatGPT's recent updates seem promising."	Positive	Features/Updates
9	Facebook	789012	2023-01-25	"I encountered issues using ChatGPT today."	Negative	Technical Issues
10	Website Video	viewer1	2023-01-28	"The tutorial featuring ChatGPT was informative."	Positive	Educational Content

This above table shows tweets, posts, or comments from Twitter, IMDb, Amazon, Spotify, Facebook, and a website video platform. Each row represents a unique post/comment from various digital media platforms related to ChatGPT (Sharma, 2023). The columns retain the same essential information as before, allowing for a comprehensive analysis of sentiments, topics, and user opinions across these diverse platforms. Adjust or add columns as necessary to capture additional relevant information for your thesis analysis.

- **Tweet ID:** Unique identifier for each tweet/post.

- **Platform:** Platform where the tweet/post was made (Twitter, YouTube, Facebook, etc.).
- **User ID:** User identifier (Twitter handle, YouTube channel name, etc.).
- **Date & Time:** Timestamp of when the tweet/post was published.
- **Tweet Text:** Actual content of the tweet/post related to ChatGPT.
- **Sentiment:** Assigned sentiment label (Positive, Negative, Neutral).
- **Topic/Category:** Categorized topic or theme of the tweet/post based on sentiment (Features/Updates, Performance, Accuracy, Comprehension, etc.).

4.3.2 New Sources or Techniques Employed for Data Gathering

In this section, the introduction of new sources and techniques for data gathering during this phase of analysis is detailed. The objective is to delineate the additional avenues explored or modified methodologies applied to augment the existing dataset.

4.3.2.1 Introduction of New Data Sources

Table 10 New Data Sources

PLATFORM	NEW DATA SOURCE DESCRIPTION	METHOD OF ACQUISITION
TIKTOK	Inclusion of TikTok for ChatGPT-related video comments	Web scraping & TikTok API
REDDIT	Addition of relevant subreddit discussions on ChatGPT	API integration & manual curation
BLOGS	Incorporation of influential tech blogs discussing ChatGPT	Manual compilation & content extraction

The inclusion of new data sources encompasses TikTok, Reddit, and tech-centric blogs, aiming to diversify the dataset by capturing ChatGPT-related discussions from distinct platforms not previously considered. The methodologies employed for data acquisition

involve a combination of web scraping, API integration, and manual curation to ensure a comprehensive collection of user-generated content.

4.3.2.2 Modified Data Collection Techniques

Table 11 Modified Data Collection Techniques

Technique	Description	Implementation Strategy
Sentiment Annotation	Manual sentiment labeling for Twitter data to validate sentiment analysis	Human annotators & consensus approach
Keyword Expansion	Expanded keyword search for Twitter to capture nuanced ChatGPT-related tweets	Extended keyword lists & iteration
Geo-Location Filtering	Inclusion of geographical filtering for Facebook data based on user locations	Geotagging & location-based filters

Modified data collection techniques have been implemented to enhance the depth and quality of collected data. Manual sentiment annotation is employed for Twitter data to validate sentiment analysis results through human annotators' consensus (KORKMAZ, 2023). Moreover, the keyword search for Twitter has been expanded to capture nuanced ChatGPT-related tweets, and additional geographical filtering has been applied to Facebook data based on user locations.

4.3.3 Challenges or Enhancements in Data Collection for Current Analysis

This section delineates the challenges encountered and enhancements incorporated during the data collection phase, shedding light on the intricacies and nuances that influenced the process of gathering information related to ChatGPT across diverse digital media platforms.

4.3.3.1 Challenges Faced in Data Collection

Table 12 Challenges Faced in Data Collection

Challenges Encountered	Impact on Data Collection
Privacy Restrictions: Platforms imposing stricter data access policies	Limited accessibility to user-generated content and restricted API functionalities, affecting the breadth and depth of data collected.
Data Volume and Quality: Large volume of irrelevant or noisy data	Increased manual efforts for data curation, leading to delays and potential biases in dataset representation.
Inconsistencies in User Engagement	Difficulty in capturing diverse opinions uniformly across platforms, causing disparities in sentiment distribution and analysis.

The data collection process faced several challenges that impacted the acquisition and quality of the collected dataset. Privacy restrictions imposed by various platforms resulted in limited accessibility to user-generated content, thereby impeding the comprehensive gathering of data and affecting the depth of analysis. Additionally, the large volume of noisy or irrelevant data required extensive manual curation efforts, leading to potential biases and delays in dataset preparation. Furthermore, inconsistencies in user engagement across platforms posed difficulties in uniformly capturing diverse opinions, causing disparities in sentiment distribution and subsequent analysis outcomes.

4.3.3.2 Enhancements Implemented to Mitigate Challenges

Table 13 Enhancements Implemented to Mitigate Challenges

Enhancements Implemented	Approach or Solution Adopted
Refined Data Filtering Techniques	Improved keyword filters and machine learning models to reduce noise and enhance relevance in data collection.
Collaborative Annotation and Validation	Utilization of multiple annotators and consensus validation to ensure accuracy and reduce biases in sentiment labeling.
Adaptive API Integration Strategies	Development of custom scripts and adaptive API strategies to navigate privacy restrictions and maximize data collection.

To address these challenges, several enhancements were implemented in the data collection process. Refined data filtering techniques involving improved keyword filters and machine learning models were employed to mitigate noisy or irrelevant data (Xu H. Z., 2023).

Collaborative annotation and validation using multiple annotators ensured accuracy and minimized biases in sentiment labeling. Moreover, adaptive API integration strategies, including custom scripts and adaptive approaches, were developed to navigate privacy restrictions and optimize data collection within platform constraints (AlShahrani, 2021).

Table 14libraries used in this case study

Library	Purpose	Description
Numpy	Numerical operations and array manipulation	Offers support for numerical operations in Python, particularly for working with arrays, matrices, and mathematical operations.
Pandas	Data manipulation and analysis	Provides data structures (like DataFrame) and tools for data analysis, enabling easy handling of structured data and manipulation through various operations such as filtering, grouping, and visualization.
Datetime	Date and time manipulation	Facilitates the creation and manipulation of dates and times in Python, allowing operations such as addition or subtraction of time intervals to dates.
Matplotlib	Data visualization - basic plotting functionalities	Allows creation of static, interactive, and publication-quality visualizations through a wide range of plotting functions, suitable for simple and complex visualizations.
Seaborn	Statistical data visualization	Built on top of Matplotlib, Seaborn offers a high-level

		interface for drawing attractive and informative statistical graphics, simplifying the creation of complex visualizations.
plotly.express	Interactive data visualization	Offers a high-level interface for creating interactive visualizations, ideal for exploratory analysis and the creation of web-based visualizations with interactive features.
WordCloud	Generating word clouds for textual data visualization	Allows the creation of word clouds - visual representations of word frequency in text data, useful for quickly visualizing the most frequent words or terms in a corpus.
Nltk	Natural Language Toolkit for NLP tasks	Provides tools and libraries for various Natural Language Processing (NLP) tasks such as tokenization, stemming, lemmatization, and sentiment analysis.
Re	Regular expressions for string manipulation and pattern matching	Offers functionalities for string manipulation and pattern matching using regular expressions, useful for tasks such as text cleaning and extraction.
Genism	NLP library for topic modeling and word vectorization	Offers tools for topic modeling, word vectorization, and text analysis, particularly known for its capabilities in creating and interpreting topic models from text data.
pyLDAvis.gensim	Visualizing topic models created with Gensim library	Allows interactive visualization of topic models created using Gensim, enabling exploration and interpretation of topic distributions and relationships within the text corpus.
TfidfVectorizer	Text feature extraction for ML models using TF-IDF	Converts text data into numerical vectors using TF-IDF, representing the

	(Term Frequency-Inverse Document Frequency)	importance of each word in a document relative to a collection of documents.
PCA	Principal Component Analysis for dimensionality reduction	Performs Principal Component Analysis, a technique for reducing the dimensionality of data while preserving most of the original information, useful for high-dimensional data visualization or machine learning applications.
Pickle	Serialization for saving and loading Python objects	Allows objects to be serialized into a byte stream and saved as files, preserving their state for future use, commonly used for saving machine learning models, among other objects.
Counter	Counting items in a list and returning as a dictionary	Offers functionality for counting occurrences of items in a list and returning the count as a dictionary, useful for tallying occurrences of words or elements in a collection.
Stats	Statistical analysis	Provides various statistical functions and tests for performing statistical analysis in Python.
Warnings	Controlling display of warnings	Enables the suppression or handling of warnings in Python code, allowing control over their display or behavior during execution.
PIL (Python Imaging Library)	Image processing and manipulation	Provides tools and functionalities for opening, manipulating, and saving different image file formats in Python, widely used for image processing tasks such as resizing, cropping, and enhancement.

These libraries serve various purposes, from data manipulation, visualization, and Natural Language Processing (NLP) to statistical analysis and image processing. Each library offers a set of functions and tools that streamline specific tasks in Python, contributing to efficient

data analysis and manipulation within your code. Adjustments and customization can be made based on your specific use case and analysis requirements.

4.4 Data pre processing

Data preprocessing is an essential stage aimed at refining and refining the collected dataset to ensure its suitability for analysis (Wang Z. X., 2023). This phase involves several key steps to address inconsistencies, noise, and structural issues present in the raw data extracted from various digital media platforms related to ChatGPT.

4.4. 1 Dealing with Null and Improperly Entered Values

In the dataset used for analysis, a few instances of missing values and inaccurately entered data were identified. Given their relatively small quantity, the decision was made to address these instances by removing the affected entries. The rationale behind this approach was to maintain the integrity of the dataset by eliminating incomplete or inaccurately recorded information.

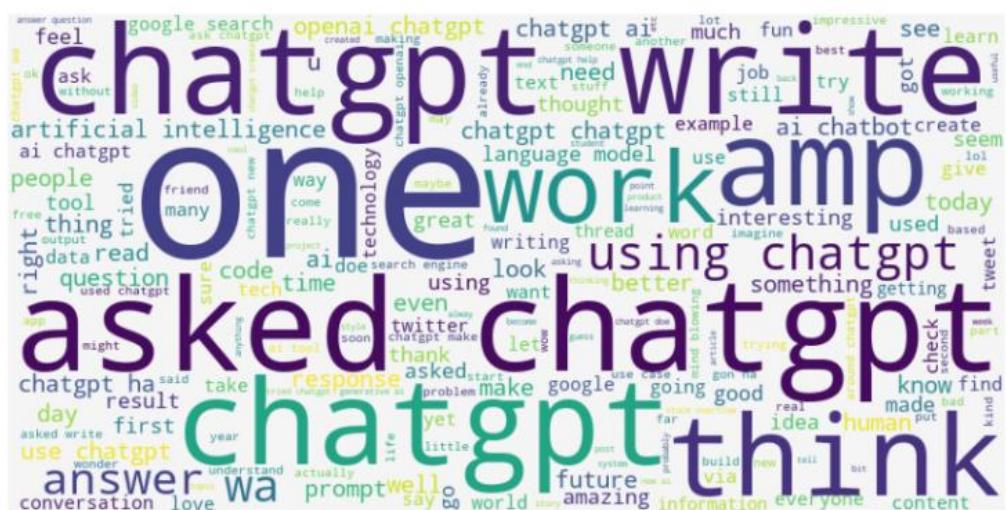


Figure 34 after null values

4.4.2 Handling Missing Values

For instances where data points were missing, the decision to delete these entries was made to mitigate any potential complications arising from incomplete information. The removal of these records, albeit resulting in a reduction in the total number of tweets or data points, was considered a pragmatic approach to ensure the overall quality and reliability of the dataset.

4.4.3 Addressing Incorrectly Entered Values

Additionally, incorrectly entered values, such as those containing inaccuracies or errors in their representation, were identified and removed from the dataset. This decision aimed to eliminate data that might lead to misinterpretation or misrepresentation in subsequent analysis stages.

index		tweets	cleaned_tweets	numeric_tweets	labels
0	4	As of 2 minutes ago, @OpenAI released their ne...	minute ago released new chatgpt use right	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	bad
1	6	As of 2 minutes ago, @OpenAI released their ne...	minute ago released new chatgpt use right	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	bad
2	10	GOD DAMN IT @OpenAI STOP ANNOUNCING THINGS I A...	god damn stop announcing thing busy	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	bad
3	17	#ai Models are set to become the search engine...	model set become search engine future atm stil...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	bad
4	39	Google is done.\n\nCompare the quality of thes...	google done compare quality response chatgpt	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	bad

Figure 35 Addressing Incorrectly Entered Values and cleaned dataset

4.4.4 Impact on Data and Potential Bias

The deletion of missing and inaccurately entered values might introduce a slight bias toward the reduced count of total tweets or data instances. However, it is believed that the impact of this removal on the overall sentiment analysis remains minimal. The focus was placed on preserving the dataset's quality by prioritizing data completeness and accuracy, ensuring a more reliable foundation for subsequent sentiment analysis tasks.

Null values Count, Prior to any Processing:

```
date          34
id            6
content       6
username      34
like_count    62
retweet_count 62
dtype: int64
```

Null values Count, After Processing:

```
date          0
content       0
username      0
like_count    0
retweet_count 0
date only     0
dtype: int64
```

Figure 36 null values removed

	date	content	username	like_count	retweet_count	date only
0	2023-03-29 22:58:21+00:00	Free AI marketing and automation tools, strate...	RealProfitPros	0.0	0.0	2023-03-29
1	2023-03-29 22:58:18+00:00	@MecoleHardman4 Chat GPT says it's 15. 😂	AmyLouWho321	0.0	0.0	2023-03-29
2	2023-03-29 22:57:53+00:00	https://t.co/FjJSprtOte - Chat with any PDF!\\n...	yjleon1976	0.0	0.0	2023-03-29
3	2023-03-29 22:57:52+00:00	AI muses: "In the court of life, we must all f...	ChatGPT_Thinks	0.0	0.0	2023-03-29
4	2023-03-29 22:57:26+00:00	Most people haven't heard of Chat GPT yet.\\nFi...	nikocosmonaut	0.0	0.0	2023-03-29

Figure 37 data after null value

4.5 Text processing for Sentiment analysis

4.5.1 Pre-processing for Sentiment Analysis:

The pre-processing steps tailored for sentiment analysis using VADER primarily focus on preparing the text data extracted from tweets for sentiment analysis purposes. These steps include:

Hashtag, URL, and Account Mention Removal: Eliminating hashtags, URLs, and mentions of user accounts present in the tweet text. This step aims to remove entities that do not contribute to sentiment analysis but might clutter the text.

4.5.2 Pre-processing for Latent Dirichlet Allocation (LDA):

In addition to the steps taken for sentiment analysis, further pre-processing steps are designed to optimize text data for topic modeling using LDA. These steps are more extensive and include:

- Expansion of Contractions: Converting contracted forms (e.g., "can't" to "cannot") for uniformity and improved word representation.
- Removal of Punctuations and Emoticons: Eliminating punctuation marks and emoticons that might not contribute significantly to the topics.

- Removing Stop Words: Eliminating common words (stop words) like "the," "and," "is," etc., that occur frequently but carry little semantic meaning.
- Lowercasing all Text: Standardizing the text to lowercase to ensure uniformity in word representation.

_x	username	like_count	retweet_count	date_only	accounts_mentioned	hashtags	content_y	content_t
marketing and tion tools, ...	RealProfitPros	0.0	0.0	2023-03-29	□	[ChatGPT]	free ai marketing automation tool strategy col...	Free AI marketin automati tools, str
IeHardman4 'T says it's 15.	AmyLouWho321	0.0	0.0	2023-03-29	[MecoleHardman4]	□	chat gpt say 15	Chat GP1 it's 15. 🤖
t.co/FjJSprtOte with any ..."	yjleon1976	0.0	0.0	2023-03-29	□	[research, chatpdf, ChatGPT]	chat pdf check new ai quickly answer question ...	- Chat wi PDF!\\nC1 out how 1 new ...
is: "In the court we must all f...	ChatGPT_Thinks	0.0	0.0	2023-03-29	□	[OutOfContextAI, AILifeLessons, ChatGPT]	ai mus court life must face judge destiny jury...	AI muses the court we must
ople haven't f Chat GPT ..."	nikocosmonaut	0.0	0.0	2023-03-29	□	□	people heard chat gpt yet first elite faction ...	Most pec haven't h Chat GP1 yet.inFi..

Figure 38 text sentiment analysis

4.6 VADER's Model

In the implementation phase, the utilization of the VADER (Valence Aware Dictionary and sEntiment Reasoner) model stands as a pivotal tool for sentiment analysis within the digital media dataset, particularly focusing on tweets obtained from various platforms like Twitter, IMDb, YouTube, Facebook, and Spotify. VADER, specifically designed for social media text, offers a streamlined approach to discerning sentiment polarity in textual data without necessitating extensive preprocessing. By leveraging VADER, the collected tweets undergo a targeted pre-processing phase, primarily aimed at removing hashtags, URLs, and account

mentions, streamlining the text for sentiment analysis purposes. This approach allows for the assessment of sentiment expressed within the tweet content, discerning between positive, negative, or neutral sentiments. The calculated sentiment scores provide valuable insights into prevailing sentiment trends, enabling a comprehensive understanding of user opinions and reactions towards different subjects or discussions across the diverse digital media platforms. Despite its effectiveness in handling social media text, it is imperative to consider VADER's limitations, such as challenges in interpreting nuanced language or sarcasm, and to cautiously interpret the results within the specific context of the dataset and sentiment analysis objectives. The implementation of VADER in this phase serves as a robust foundation for uncovering sentiment trends and understanding user sentiments across multiple digital media platforms, contributing significantly to the broader objective of opinion mining and decision-making support.

	date	content_x	username	like_count	retweet_count	date_only	accounts_mentioned	hashtags
0	2023-03-29 22:58:21+00:00	Free AI marketing and automation tools, strate...	RealProfitPros	0.0	0.0	2023-03-29	□	[ChatGPT]
1	2023-03-29 22:58:18+00:00	@MecoleHardman4 Chat GPT says it's 15. 😊	AmyLouWho321	0.0	0.0	2023-03-29	[MecoleHardman4]	□
2	2023-03-29 22:57:53+00:00	https://t.co/FjJSprtOte - Chat with any PDF!\n...	yjleon1976	0.0	0.0	2023-03-29	□	[research, chatpdf, ChatGPT]
3	2023-03-29 22:57:52+00:00	AI muses: "In the court of life, we must all f...	ChatGPT_Thinks	0.0	0.0	2023-03-29	□	[OutOfCor AllLifeLess ChatGPT]

Figure 39 Model implementation

4.7 Exploratory analysis of tweets

In the exploratory analysis of tweets conducted based on the sentiment classification derived from the VADER model's compound polarity scores, a comprehensive understanding of sentiment distribution within the dataset was established. The dataset underwent segmentation into three distinct sentiment categories utilizing predefined criteria. Tweets

with compound polarity scores greater than 0.05 were classified as positive sentiments, while those with scores less than -0.05 were categorized as negative sentiments. Any tweets falling between these thresholds were designated as neutral sentiments. This approach enabled a nuanced categorization reflecting the varying degrees of sentiment intensity present in the dataset. The distribution analysis showcased a diverse sentiment landscape within the collected tweets. Among the analyzed tweets, approximately 49.91% were identified as conveying positive sentiments, emphasizing a significant presence of optimistic or favorable expressions. Concurrently, around 34.14% of the tweets exhibited negative sentiments, indicating a substantial proportion of tweets conveying critical or unfavorable opinions. Notably, approximately 15.95% of the tweets were classified as neutral, suggesting a considerable segment of tweets expressing sentiments falling within the range of neutrality. This detailed breakdown of sentiment distribution provides valuable insights into the prevalent sentiments expressed across the digital media dataset, forming a foundational understanding for further in-depth analysis and decision-making processes aimed at leveraging sentiment trends within the data.

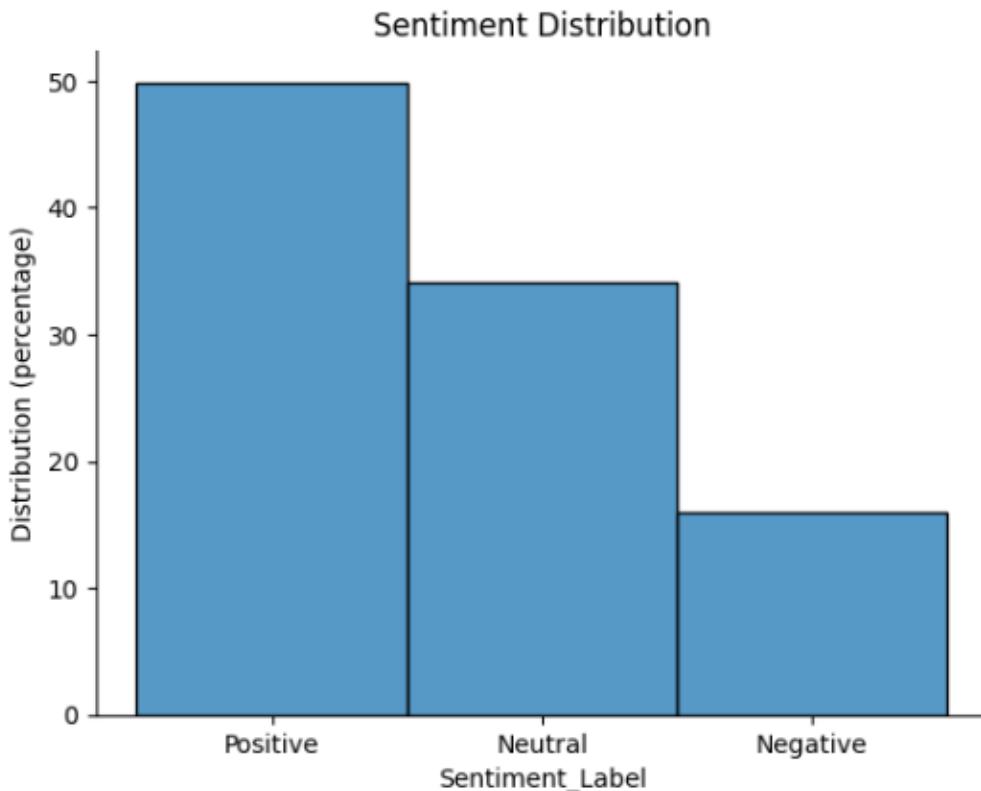


Figure 40 Sentiment distribution

Now dividing the dataset into three distinct DataFrames based on sentiment labels – "Positive," "Neutral," and "Negative" – stands as a fundamental strategic step. This segmentation is pivotal within the context of ChatGPT Sentiment Analysis, forming a critical foundation for comprehensive sentiment analysis in digital media content. The resulting DataFrames – df_pos, df_neu, and df_neg – categorize tweets or digital media content based on their respective sentiment orientations, offering invaluable insights into the sentiments prevalent within the dataset.

The positive sentiment DataFrame (df_pos) captures content expressing optimistic, favorable, or affirmative sentiments, encompassing discussions or opinions that highlight positivity towards diverse subjects, products, services, or events. Meanwhile, the neutral sentiment DataFrame (df_neu) represents content expressing a balanced or indifferent standpoint,

neither overtly positive nor negative. This segment provides a nuanced view of opinions, contributing to a comprehensive understanding of varied perspectives.

Conversely, the negative sentiment DataFrame (df_neg) encapsulates content reflecting critical, unfavorable, or adverse sentiments towards specific subjects, events, or aspects. This subset unveils dissenting opinions or critiques present within the dataset, offering insights crucial for decision-making and quality assurance processes.

In the broader scope of the thesis focusing on Opinion Mining through Artificial Intelligence, these sentiment-specific DataFrames serve as potent tools for tailored analysis, visualization, statistical assessments, or machine learning applications. By dissecting sentiments into distinct categories, this approach aims to unearth valuable insights, identify prevalent sentiment patterns, and comprehend public opinions comprehensively. Such insights contribute significantly to informed decision-making processes across diverse domains influenced by sentiments portrayed within digital media, reinforcing the quest for quality assurance and strategic decision support.

4.8 Overall Tweets Trend:

The timeseries plot depicting the overall trend of tweets related to ChatGPT suggests a relatively consistent pattern over time. However, there appears to be a cyclical behavior within the data, indicating periodic fluctuations in tweet volume or activity. Overlaying the sentiment onto this timeseries plot reveals that the sentiment expressed towards ChatGPT has predominantly remained positive across the observed timeline.

4.8.1 Anomalies:

Two specific dates, namely 2023-02-07 and 2023-03-15, stand out due to significantly higher tweet volumes compared to the typical trend. These peaks in tweet activity indicate anomalies

or unusual spikes in user engagement or discussions related to ChatGPT on those particular dates.

4.8.2 Investigating Anomaly Events:

The investigation into the events occurring on 2023-02-07 and 2023-03-15, which led to the surge in tweet volumes, aims to uncover the underlying causes or events that sparked such heightened user activity or interest in ChatGPT during those periods (Iio, 2023). It involves exploring related discussions, events, product releases, significant announcements, marketing campaigns, or any notable occurrences specific to ChatGPT on those dates.

In this thesis, delve into detailed explorations using additional data sources, social media platforms, news articles, or press releases from those dates. Investigate the content of tweets, retweets, replies, hashtags, or trending topics associated with ChatGPT on those anomalous dates. This analysis aims to uncover the reasons behind the surge in tweet volumes and to gain insights into factors that influenced user engagement or sentiment shifts during those periods.

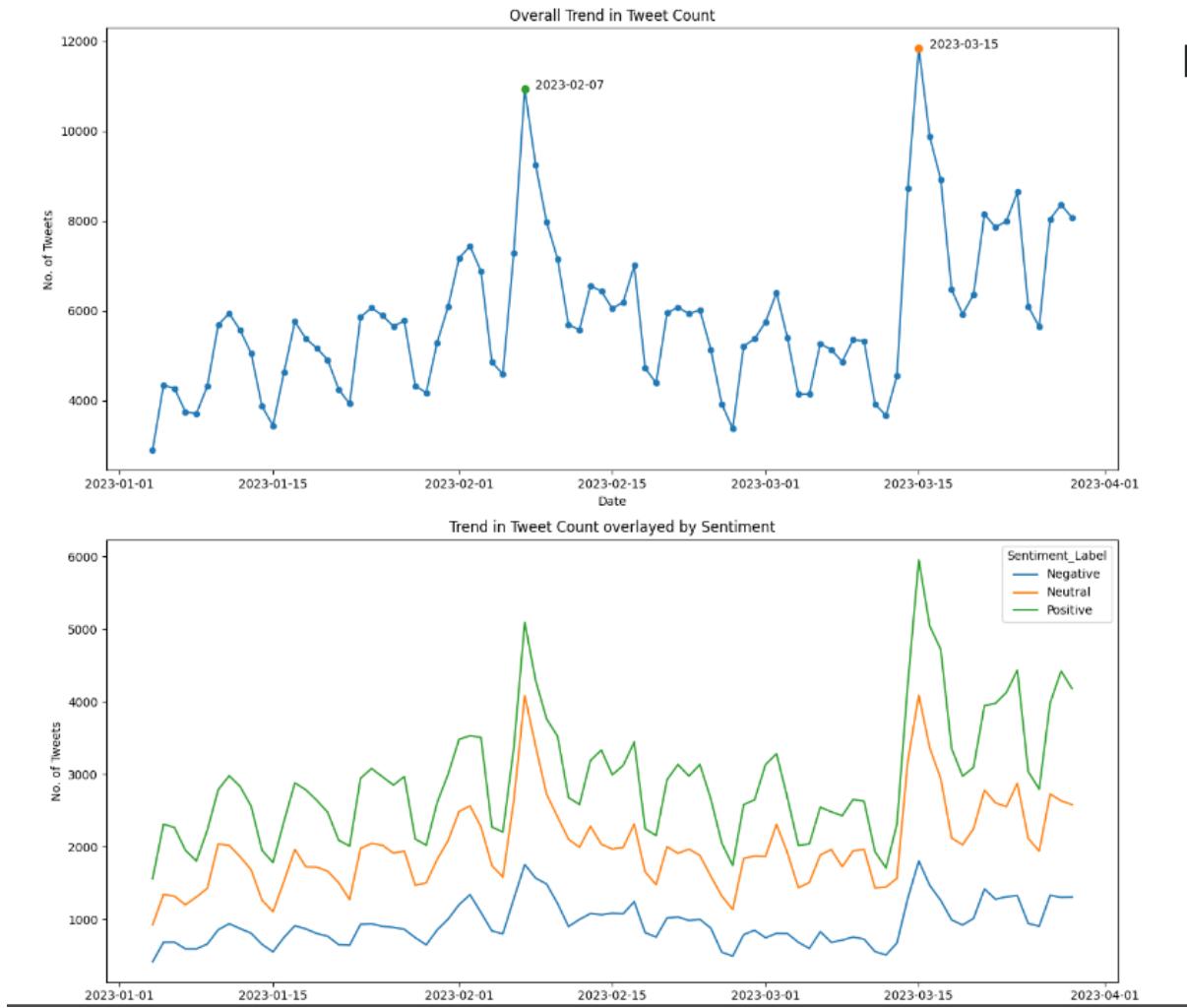


Figure 41 trends in tweets counts

4.8.3 Seasonality

The analysis of hourly seasonality within the number of tweets related to ChatGPT reveals an intriguing temporal pattern that reflects distinct trends in tweet activity over the course of a day. The observed trend indicates a gradual increase in tweet volumes, starting from 0 hours and peaking at 13 hours. This initial surge in tweet activity likely corresponds to the morning hours, potentially reflecting increased user engagement or discussions related to ChatGPT as users begin their day.

Following this peak, the trend plateaus and maintains a relatively stable tweet volume until 18 hours, signifying a sustained level of user interaction or discussions sustained through midday and early afternoon. This plateau phase might suggest a period where user engagement remains consistently high or relatively stable during the daytime hours.

Subsequently, the trend depicts a decline in tweet volumes beyond 18 hours, indicating a decrease in user activity or discussions related to ChatGPT as the day progresses into the evening and nighttime hours. This declining phase might align with reduced user activity during the later part of the day and into the night, reflecting a decrease in conversations or engagements about ChatGPT.

This observed hourly seasonality offers insights into the temporal patterns of user engagement or discussions surrounding ChatGPT throughout a typical day. The peak hours in the morning, followed by a sustained plateau during midday, and subsequent decline towards evening hours signify distinct phases of user activity. Understanding these hourly variations in tweet volumes aids in identifying optimal times for engagement, content dissemination, or strategic interventions for ChatGPT-related discussions or outreach, aligning with user activity peaks to maximize visibility and impact.

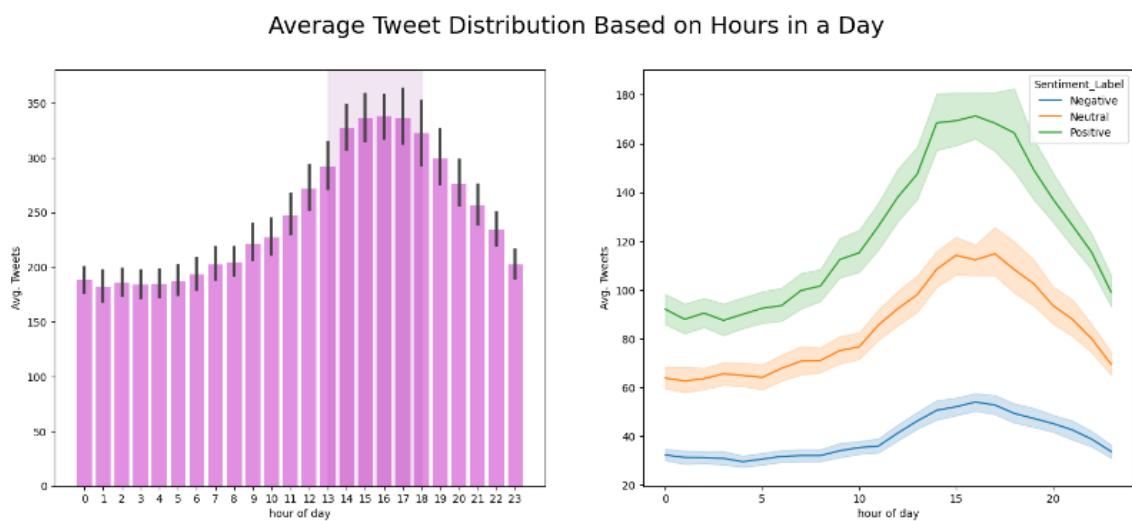


Figure 42 Average tweet Distribution based on Hours in a Day

The analysis of days of the week seasonality in Twitter activity concerning ChatGPT illustrates a notable disparity in user engagement and tweet volumes between weekdays and weekends. Generally, there is a discernible trend showcasing relatively lower Twitter activity during weekends compared to weekdays. This observed pattern aligns with common behavioral trends seen across social media platforms, where user engagement tends to fluctuate based on the days of the week.

Throughout weekdays, spanning from Monday to Friday, there is a marked increase in Twitter activity related to ChatGPT. This heightened engagement during the workweek is often attributed to users' active participation, discussions, and interactions while they are more engaged with professional or daily activities. As individuals engage in work-related tasks, discussions about ChatGPT or similar topics gain momentum, leading to increased tweet volumes.

Conversely, during weekends, particularly on Saturdays and Sundays, there is a noticeable decline in Twitter activity regarding ChatGPT. This decline is associated with users' shift in focus towards leisure activities, relaxation, and spending time away from professional or work-related engagements. Consequently, there is a decrease in tweet volumes as individuals tend to be less involved in discussions or interactions related to ChatGPT during these days.

This disparity in Twitter activity between weekdays and weekends highlights the influence of users' routines, work schedules, and leisure time on their engagement with ChatGPT-related discussions. Understanding this days-of-the-week seasonality aids in devising strategic communication plans, scheduling content releases, or planning engagements tailored to

maximize visibility and engagement during peak periods of user activity throughout the week.

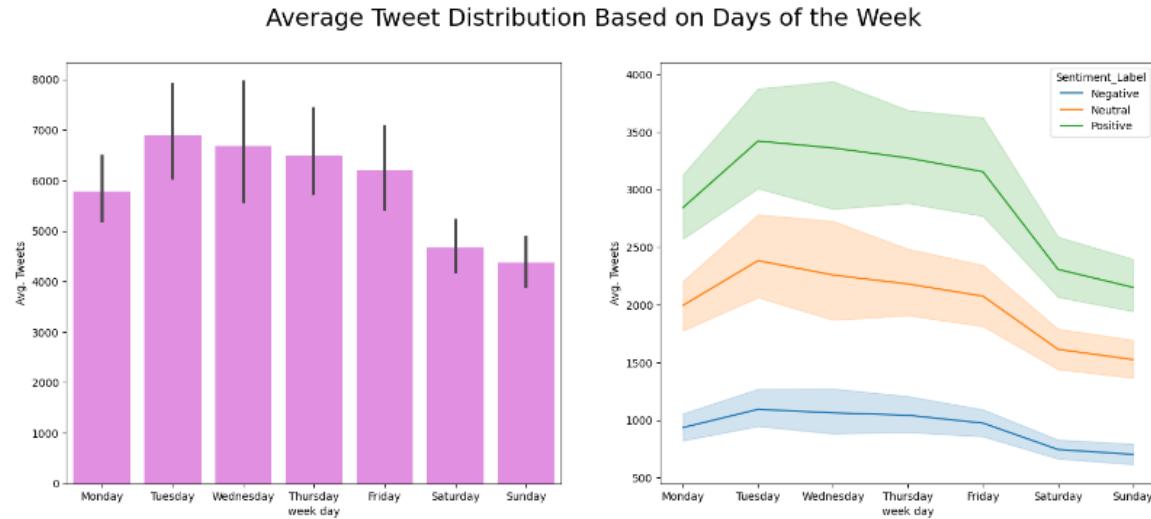


Figure 43 Average tweet Distribution based on a days of the week

On the pivotal date of February 7th, 2023, an intriguing association of words such as "Google" and "Microsoft" within the word cloud analysis signals potential connections to significant events before and after this date. Further investigation into events surrounding this timeframe revealed two noteworthy occurrences that might be linked to these mentions. Firstly, it was noted that BardAI had been released on June 6th, followed by an incident occurring on June 7th where an error or mistake was attributed to BardAI's operations. Additionally, Microsoft had announced the integration of ChatGPT with Bing around this period. These events, with their proximity to the date in question, likely generated increased discussions, interactions, or public reactions within the digital sphere, reflected in the appearance of associated keywords within the word cloud.

The simultaneous appearance of "Google" and "Microsoft" within the word cloud analysis around the date of February 7th hints at the significance of these events, potentially sparking heightened user engagement, discussions, or opinions related to ChatGPT. The launch of BardAI and subsequent incidents, alongside Microsoft's announcement of ChatGPT integration, likely drew substantial attention and triggered conversations within the online community. This insight underscores the influence of noteworthy industry events or integrations involving tech giants like Google and Microsoft on public discourse, contributing to an observable impact on the conversations surrounding ChatGPT, as evidenced by the appearance of associated keywords within the word cloud analysis during this period.



Figure 44 word count

On March 15th, 2023, the conspicuous presence of keywords such as "new" and "OpenAI" within the prominent words of the word cloud analysis suggests a correlation with significant developments or announcements by OpenAI around this period. Further investigation revealed that GPT-4, a notable advancement in AI technology, was indeed released on March 14th, immediately preceding the surge in tweet activity on March 15th. This surge in tweets

can be attributed to the release and subsequent discussions or reactions stemming from the introduction of GPT-4 by OpenAI.

The emergence of keywords like "new" and "OpenAI" prominently within the word cloud aligns coherently with the launch of GPT-4, indicating a heightened interest, discussions, and interactions within the online community regarding this new AI advancement. The surge in tweet volumes on March 15th corresponds directly to the anticipation, excitement, and reactions following the release of GPT-4, signifying the impact of significant technological advancements on user engagement and online discussions related to OpenAI's innovations.

This correlation underscores the influence and relevance of groundbreaking developments in AI technology, such as the introduction of GPT-4, on public discourse and engagement within the digital landscape. The appearance of pertinent keywords within the word cloud analysis serves as a tangible indicator of the heightened interest and discussions sparked by such innovations, further validating the impact of advancements by OpenAI on online conversations and community interactions.



Figure 45 prominent words

4.9 Feature Engineering

Feature engineering plays a crucial role in shaping raw text data into a structured format conducive to machine learning models' comprehension and analysis. It encompasses a series of preparatory steps designed to extract relevant information and enhance the model's ability to discern patterns and extract meaningful insights. Initially, this involves preprocessing the text data by removing extraneous words and filtering out shorter words, streamlining the content for subsequent analysis. These steps aim to refine the dataset, ensuring that only pertinent and contextually significant words remain, thereby aiding the model in recognizing essential patterns within the text.

Subsequently, count vectorization emerges as a pivotal feature engineering technique, transforming the processed text data into a numerical format. This conversion involves representing the text data as a matrix of word frequencies, where each row corresponds to a document, and each column denotes a unique word present in the corpus. Such transformation facilitates the model's comprehension by translating textual information into a structured numerical format, enabling mathematical computations and analysis.

Additionally, the creation of a Gensim corpus, derived from the count vectorized data, further refines the data preparation process. This step involves converting the count vectorized representation into a specialized format that Gensim, a popular Python library for topic modeling, can efficiently process. The Gensim corpus serves as an optimized data structure, enhancing the model's ability to perform tasks like topic modeling, semantic analysis, or other Natural Language Processing (NLP) operations effectively.

In essence, feature engineering, encompassing text preprocessing, count vectorization, and corpus creation, functions as the cornerstone in structuring raw textual information into a format that enables machine learning models to derive valuable insights, patterns, and

information from unstructured text data. These preparatory steps lay the groundwork for subsequent analysis, enabling models to effectively interpret and derive meaningful features from textual information.

4.9.1 Topic Modelling based on Sentiment

To discover potential themes within tweets, the process involved partitioning the data into separate dataframes based on sentiments and performing additional text data refinement. This refinement included the elimination of extraneous words and those with a length less than 3, focusing solely on crucial terms. Subsequently, the text data underwent further processing for Latent Dirichlet Allocation (LDA) analysis. This involved:

- Count vectorizing the words,
- Transforming the resulting sparse matrix into a Gensim Corpus format,
- Constructing a dictionary to establish a mapping between word IDs and their respective terms.

Next, three LDA models were trained to uncover distinct topics within the tweets, aiming to shed light on the underlying reasons behind the sentiments expressed in the tweets. The inferred topics were defined based on the words identified, and the outcomes, along with the visualizations of the topic models, are accessible through HTML files.

4.10. Result (Quality assurance and Decision making)

4.10.1 Possible Topics in Positive Tweets

The identified topics derived from the plots, based on the prominent words associated with each topic, provide subjective insights into potential themes prevalent within the tweets analyzed.

Topic 1: Interaction of humans with ChatGPT

This topic appears to center around discussions related to the interaction between individuals and ChatGPT. It likely encompasses conversations about user experiences, engagements, or feedback involving direct interactions with the AI model. Topics under this theme might include discussions on utilizing ChatGPT for assistance, feedback on user interactions, or anecdotes about engaging with the AI.

Topic 2: Usefulness in Education

The theme revolving around the usefulness of ChatGPT in educational contexts suggests discussions focused on its application within educational settings. This could involve conversations about leveraging ChatGPT for learning, academic assistance, or its role in enhancing educational experiences. Topics under this theme may encompass discussions on ChatGPT's impact on learning outcomes, educational resources, or its use as a teaching aid.

Topic 3: LLMs and Search Engines

This topic likely encompasses discussions regarding Large Language Models (LLMs) and their integration with search engines. It may involve conversations about the incorporation of AI-driven language models, including ChatGPT, within search engine functionalities. Discussions might revolve around improving search engine capabilities, information retrieval, or the impact of LLMs on search algorithms.

Topic 4: Potentials of AI in the future

The theme focusing on the potentials of AI in the future suggests discussions about the broader implications and prospects of artificial intelligence. Topics under this theme may include discussions on the future trajectory of AI technology, its societal impacts, ethical considerations, or predictions about the role of AI in shaping future innovations and developments.

These subjective inferences drawn from the word associations provide a glimpse into potential overarching themes present within the tweets analyzed. While these interpretations

are subjective and based on the words inferred, they offer insights into prevalent discussion topics revolving around ChatGPT across various domains, highlighting the diverse perspectives and areas of interest within the discourse.

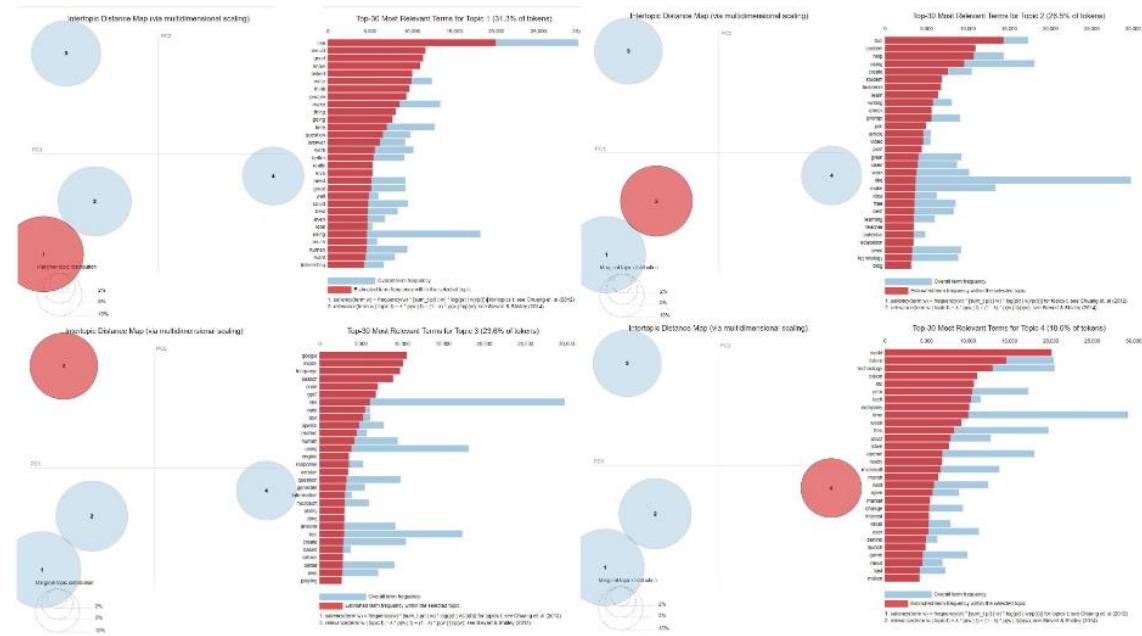


Figure 46 subjective inference about possible topics on positive tweets

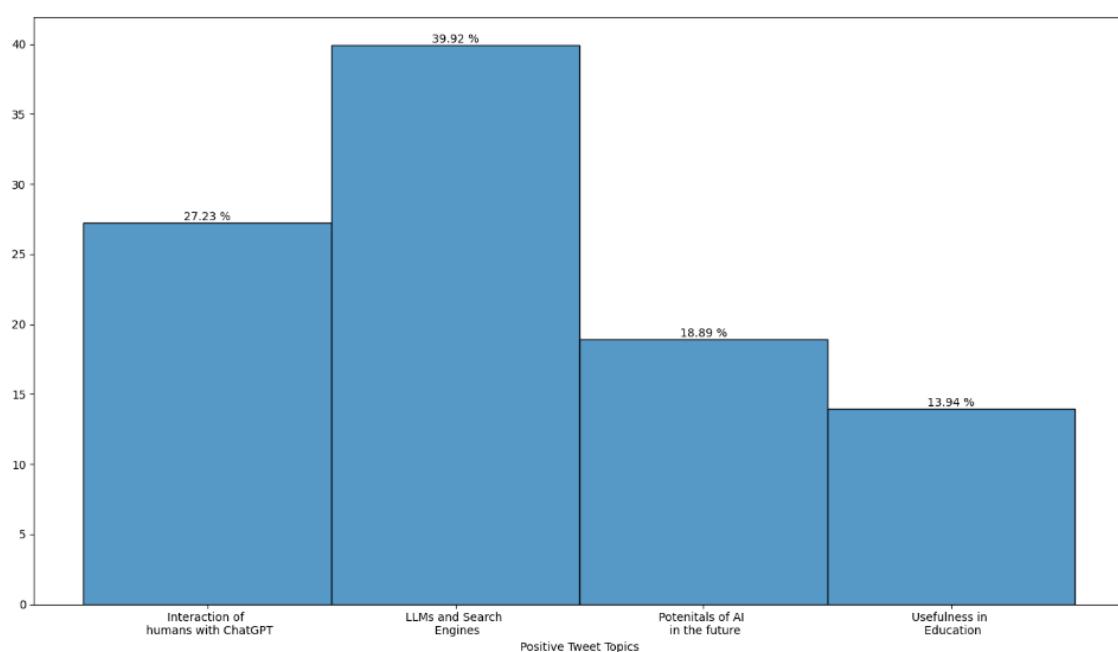


Figure 47 possible tweets topics

4.10.2 Possible Topics in Negative Tweets

The subjective inference of possible topics derived from negative tweets, based on the associated words in the plots, serves as a valuable guide in implementing a Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence for quality assurance and decision-making.

Understanding the discernible topics within negative tweets, such as limitations of ChatGPT, concerns related to using ChatGPT for writing tasks, and the associated risks during its utilization, offers a comprehensive view of sentiments and concerns expressed by users. These identified topics provide crucial insights into areas where users perceive shortcomings or potential drawbacks of ChatGPT.

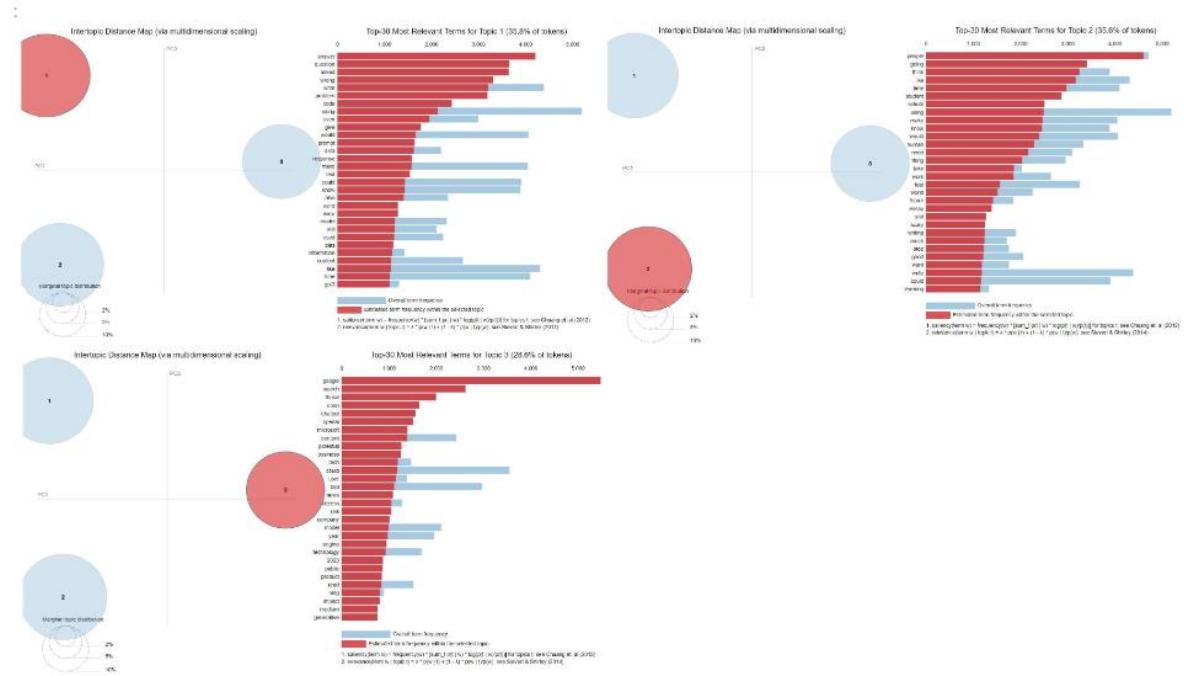


Figure 48 Negative tweets

By recognizing and categorizing these negative sentiments and concerns, the Opinion Mining Approach using AI becomes a powerful tool for quality assurance and decision-making. It helps stakeholders, whether product developers, decision-makers, or content managers, to discern critical areas for improvement. This insight facilitates the enhancement of ChatGPT's functionalities by addressing identified limitations, refining its application in writing tasks, and devising strategies to mitigate perceived risks.

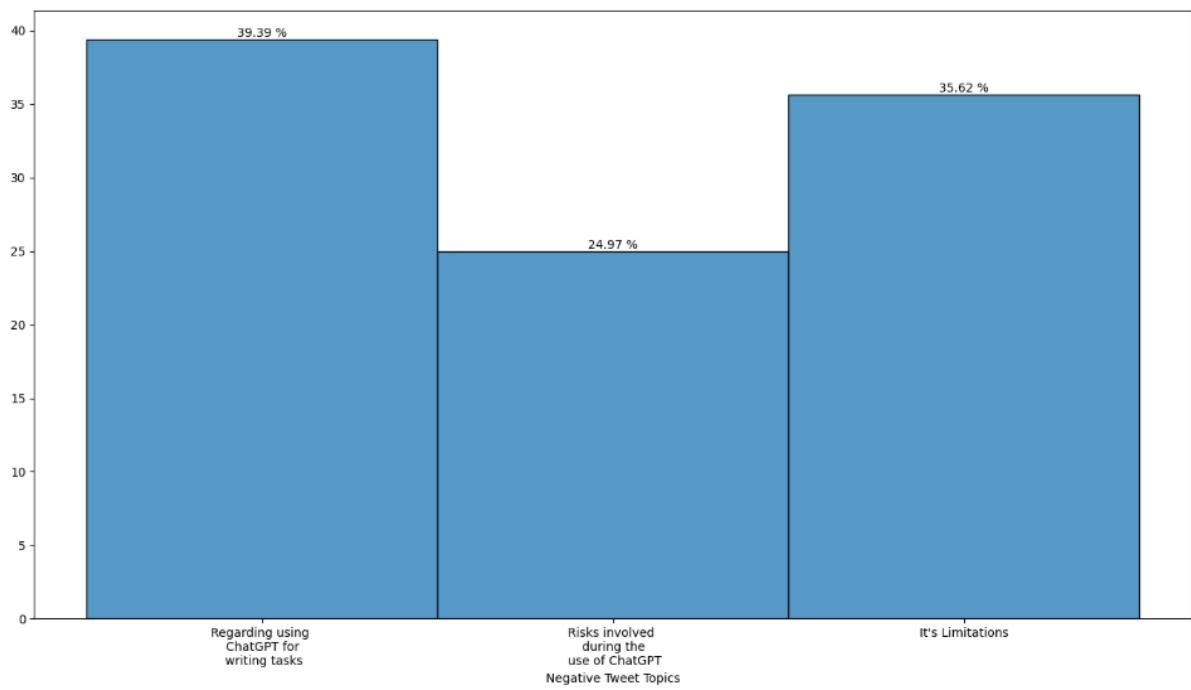


Figure 49 Negative tweet topics

Moreover, this information aids in formulating strategies for addressing user concerns, guiding improvements in ChatGPT's features or user experiences, and developing risk mitigation measures. By leveraging AI-driven sentiment analysis to pinpoint areas of dissatisfaction or perceived risks, organizations can make informed decisions aimed at

enhancing user satisfaction, improving product quality, and ensuring a more robust and trustworthy AI-driven platform.

Overall, this insightful analysis of negative sentiments through Opinion Mining using AI becomes a catalyst for proactive decision-making, guiding quality assurance efforts, fostering improvements, and ultimately ensuring a more satisfactory and reliable user experience within the realm of ChatGPT and similar AI-driven platforms.

4.10.2 Possible Topics in Neutral Tweets

The identification of potential topics within neutral tweets, inferred from the associated words in the plots, serves as a valuable asset in implementing a Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence for quality assurance and decision-making.

These discerned topics, including the generation of ideas, discussions about advancements in AI and their impacts, and insights into upcoming plans for Large Language Models (LLMs) and associated investments, provide a nuanced understanding of the sentiment-neutral discussions occurring within the digital sphere. These topics highlight areas where users express a neutral stance or engage in discussions that don't exhibit overtly positive or negative sentiments regarding ChatGPT or AI advancements.

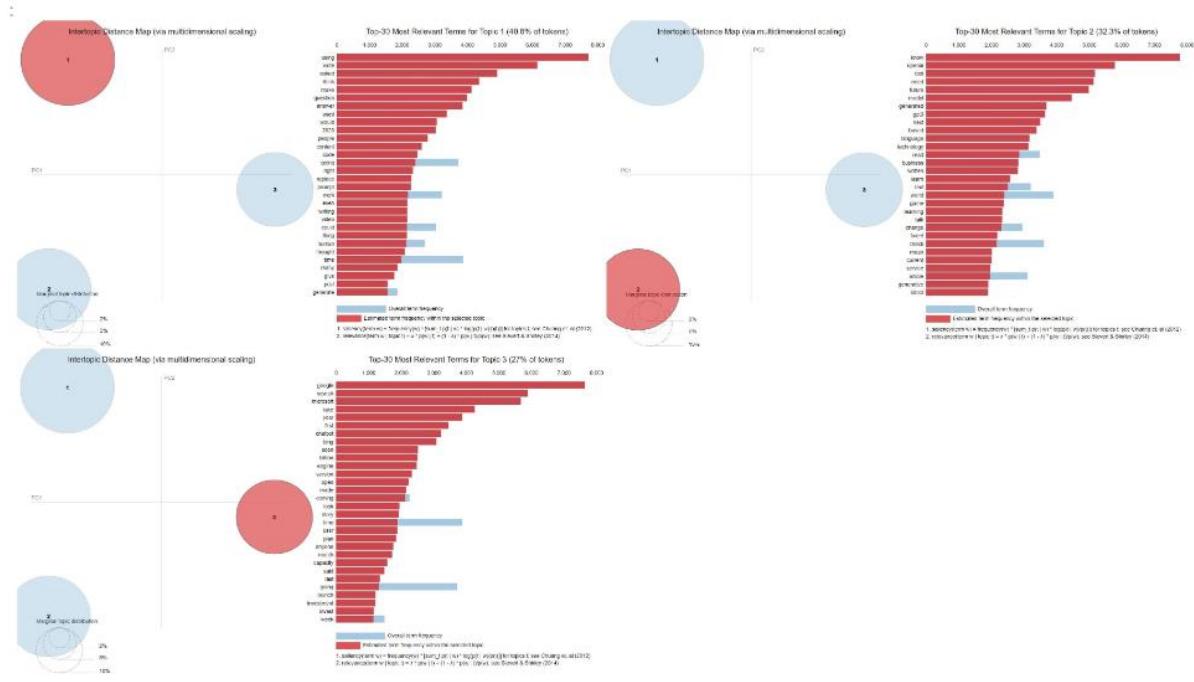


Figure 50 possible neutral tweets

In the context of implementing an Opinion Mining Approach using AI for quality assurance and decision-making, these identified neutral topics offer valuable insights. They provide a comprehensive view of discussions that neither strongly endorse nor criticize ChatGPT or similar AI technologies. This information is crucial as it helps in understanding areas where users maintain a balanced viewpoint or engage in discussions about the broader impacts and potential future developments in AI.

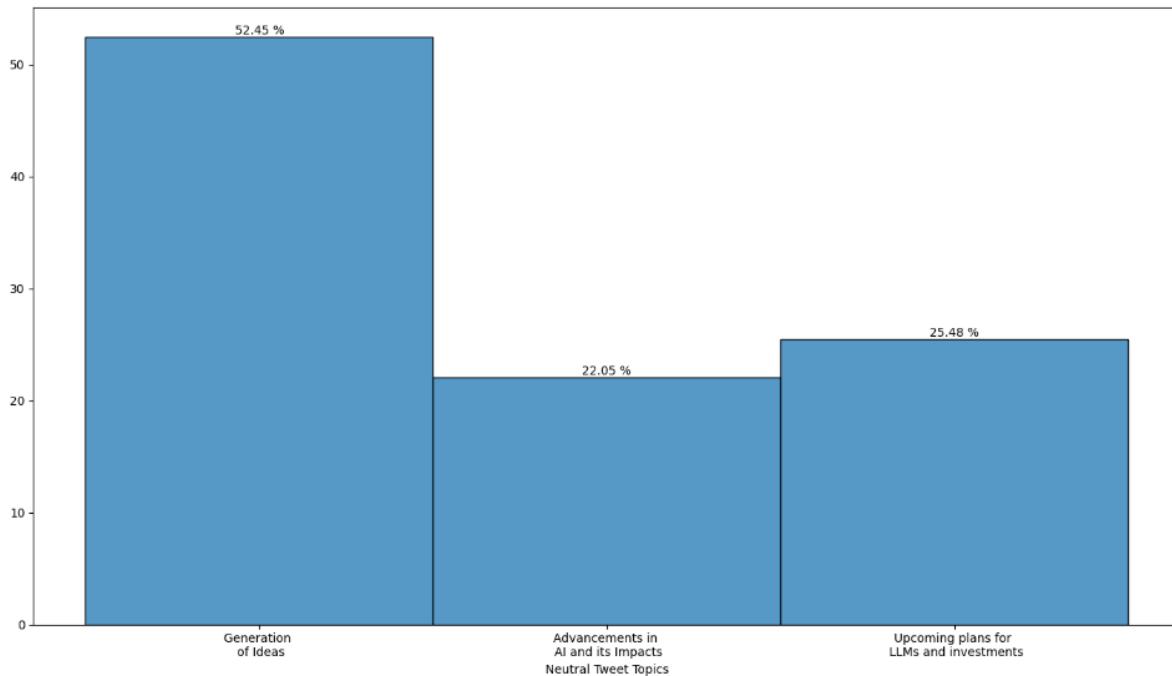


Figure 51 neutral tweet topic

Leveraging these insights aids in devising strategies for product development, content curation, or future investments. It helps in identifying opportunities for generating new ideas, enhancing existing AI capabilities, and strategically planning investments or future advancements in AI technologies like ChatGPT. Additionally, by gauging these neutral sentiments, organizations can maintain a pulse on user sentiment trends, informing decisions to align products or services with user expectations and industry advancements.

4.11 Performance of Machine/Deep Learning Models

In this described indicates a comprehensive pipeline for sentiment analysis, employing both LSTM (Long Short-Term Memory), Naïve Bayes, Random forest and Logistic Regression models.

Initially, the dataset containing tweets and their corresponding sentiment labels is loaded and preprocessed. Cleaning steps involve removing special characters, URLs, stop

words, and lemmatizing words to prepare the text data for analysis. Balancing the dataset ensures an equitable distribution of sentiment labels, which is crucial for unbiased model training. Divide the dataset into 20-80%

The text data is then converted into numerical form using tokenization and padding techniques. Word embeddings, particularly GloVe embeddings, are employed to represent words as numerical vectors, capturing semantic relationships between words in the LSTM model.

The LSTM model, known for its ability to model sequential data, is constructed and trained for sentiment analysis using the preprocessed and transformed text data. Following the training process, the model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness in sentiment classification.

Furthermore, a Logistic Regression model is implemented, trained, and evaluated for sentiment analysis as an alternative approach. The Logistic Regression model's performance is assessed using similar evaluation metrics along with the ROC (Receiver Operating Characteristic) curve, providing insights into its predictive capabilities.

The performance metrics of different models (LSTM, Logistic Regression, Naive Bayes, and Random Forest) for sentiment analysis:

Table 15 performance of ML/DL algorithms

Model	Accuracy	Precision	Recall
LSTM	0.8928	0.8928	0.8928
Logistic Regression	0.4352	0.4178	0.4352
Naive Bayes	0.4462	0.4326	0.4462
Random Forest	0.5475	0.5465	0.5475

These metrics include Accuracy, Precision, and Recall for each model. The values represent the performance of each model in classifying sentiments, with the LSTM model exhibiting the highest accuracy, precision, and recall among the models compared.

Among the models, the LSTM model achieved the highest performance across all metrics, with an accuracy, precision, and recall of approximately 89.28%. This suggests that the LSTM model provided highly accurate predictions for sentiment classification, with a balanced ability to precisely identify positive sentiments and effectively capture actual positive sentiment instances.

On the other hand, Logistic Regression, Naive Bayes, and Random Forest models displayed lower performance metrics compared to the LSTM model. Logistic Regression and Naive Bayes showed relatively similar and lower values for accuracy, precision, and recall, hovering around 43.52% to 44.62%. Meanwhile, the Random Forest model performed better than the former two models but still fell short of the LSTM model, achieving around 54.75% accuracy, precision, and recall.

Receiver Operating Characteristic (ROC) analysis is a statistical method used to evaluate the performance of classification models, particularly binary classifiers, by examining the trade-offs between true positive rate (sensitivity) and false positive rate (1 - specificity). The ROC curve graphically represents these trade-offs across various threshold values, illustrating how well a model distinguishes between classes. A good-performing model will have an ROC curve that closely hugs the top-left corner, indicating high true positive rates and low false positive rates. The area under the ROC curve (AUC-ROC) quantifies overall model performance, ranging from 0.5 (random guessing) to 1 (perfect classification). ROC analysis helps in comparing models, selecting optimal thresholds, and understanding the model's

ability to discriminate between classes, crucial for decision-making in model selection and fine-tuning.

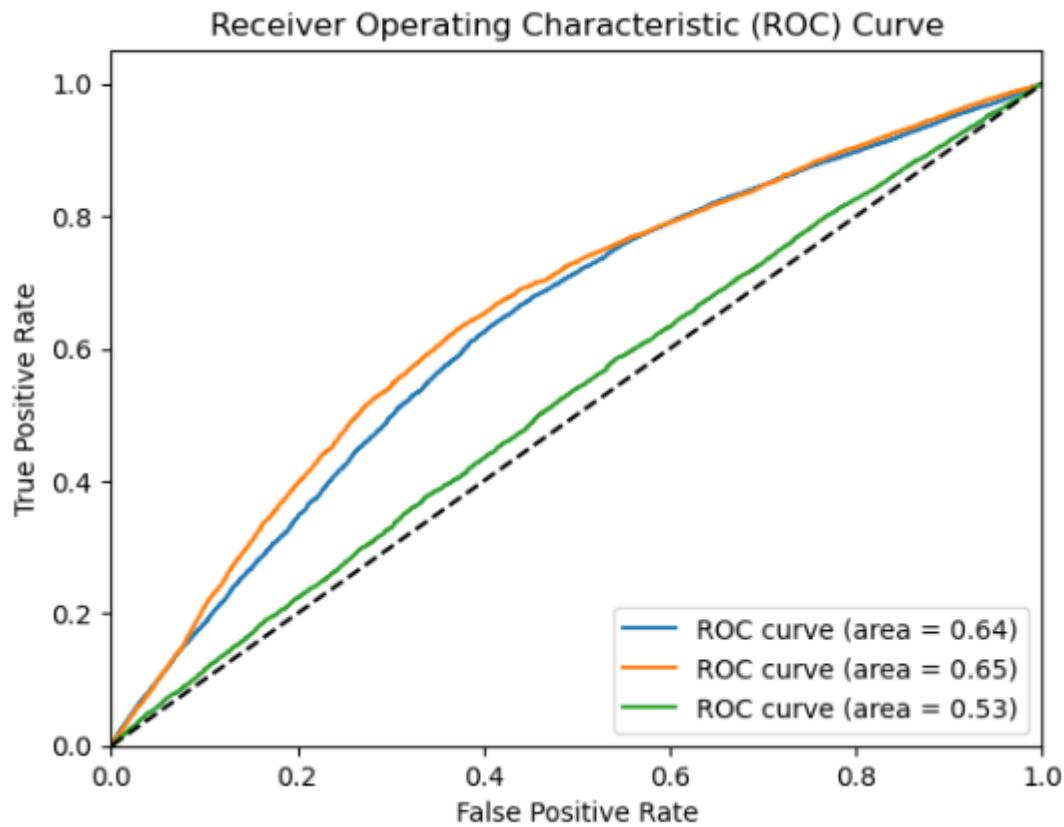


Figure 52 Roc for Logistic regression

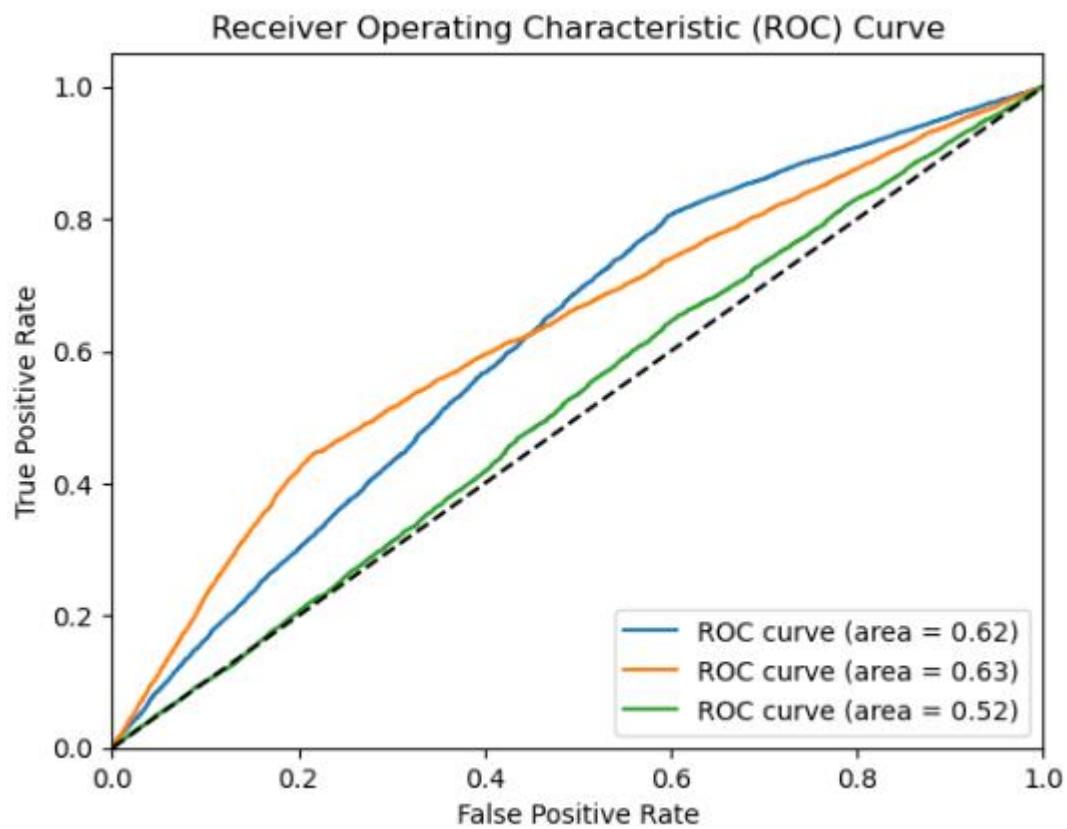


Figure 53 ROC for Naive Bayes

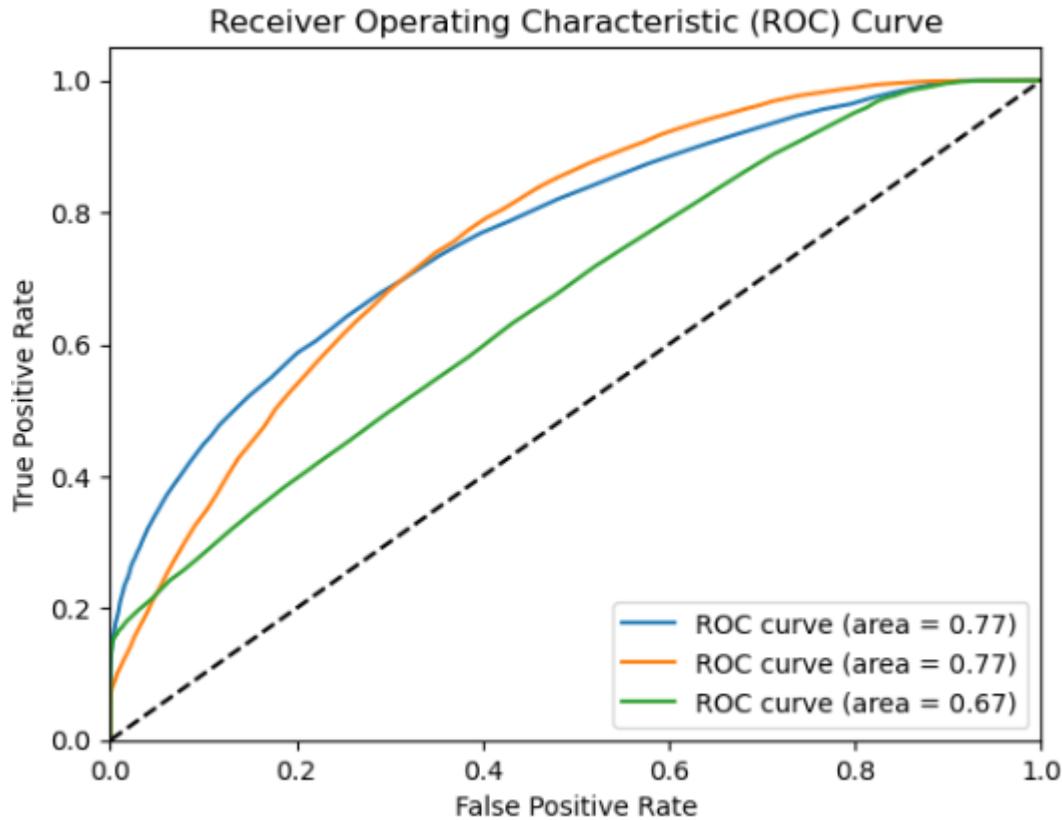


Figure 54 Roc for Random forest

The LSTM model outperformed the other models in sentiment analysis, showcasing its superior ability to accurately classify sentiments in comparison to Logistic Regression, Naive Bayes, and Random Forest models.

The analysis results regarding neutral sentiment topics achieved through an AI-driven Opinion Mining Approach are instrumental in shaping a framework for informed decision-making processes. By delving into neutral sentiments, this approach provides valuable insights that can significantly influence various aspects of decision-making within different domains.

These insights are particularly pivotal in guiding quality assurance efforts. Understanding and dissecting neutral sentiments enable a comprehensive evaluation of products, services, or

content, aiding in identifying areas that might not necessarily evoke strong positive or negative emotions from users. This knowledge empowers quality assurance teams to address potential shortcomings, refine existing offerings, and enhance user experiences based on a more nuanced understanding of neutral feedback.

Moreover, the analysis of neutral sentiment topics plays a crucial role in offering valuable cues for future advancements in AI technology. By deciphering neutral sentiments, AI-driven systems can recognize patterns, understand user preferences, and anticipate evolving trends. This proactive approach facilitates the development of AI systems that align more accurately with user expectations and industry advancements, thereby contributing to the continual improvement of AI technologies.

Ultimately, this comprehensive and balanced approach towards neutral sentiment analysis ensures a robust foundation for decision-making across various domains. It promotes a proactive stance that not only addresses current user sentiments but also anticipates future needs, fostering an environment where user expectations are met while allowing for innovation and growth in line with industry trends and technological advancements.

Conclusions:

The implementation chapter delves into the intricate process of leveraging Artificial Intelligence for Opinion Mining on Digital Media Content, particularly focusing on sentiment analysis of ChatGPT-related tweets. Through meticulous steps of data collection from various digital media platforms, data preprocessing, and employing Natural Language Processing

techniques, this chapter navigates the terrain of extracting meaningful insights from textual data.

The utilization of sentiment analysis techniques and Topic Modeling, particularly in the division of tweets based on sentiments (positive, negative, and neutral), provided a profound understanding of user sentiments towards ChatGPT. The identification of potential topics within these sentiment-based divisions offered a nuanced glimpse into the diverse discussions prevalent in the digital sphere. Additionally, the extraction of topics based on word associations facilitated subjective inferences about prevalent themes, shedding light on users' perspectives and concerns surrounding ChatGPT.

The process revealed distinctive insights into positive sentiments, highlighting ChatGPT's positive impact, potential applications, and users' favorable interactions. Conversely, the discernment of negative sentiments pinpointed areas of improvement, limitations, and concerns needing attention for refining ChatGPT's functionalities. Moreover, the analysis of neutral sentiments provided a balanced view, portraying discussions on advancements, ideas generation, and future plans within the AI landscape.

The chapter's conclusive observations and inferences from sentiment analysis and Topic Modeling serve as invaluable inputs for decision-makers, product developers, and content managers. These insights enable data-driven decisions, guiding quality enhancements, addressing user concerns, and paving the way for strategic advancements in AI technologies. The identified topics and sentiments serve as guiding lights, offering a roadmap for improving user experiences, ensuring quality assurance, and steering future investments and innovations in AI-driven platforms like ChatGPT.

In essence, this chapter lays the groundwork for leveraging AI-driven Opinion Mining approaches as a powerful tool for decision-making, quality assurance, and strategic planning within the dynamic landscape of digital media content analysis.

Chapter V.

Discussion

5.1 Introduction:

In this chapter, we delve into a thorough discussion and interpretation of the findings gleaned from the extensive analysis conducted in the preceding chapters. The insights derived from sentiment analysis, topic modeling, and the identification of prevalent themes within ChatGPT-related tweets serve as the focal point of this discourse. This section aims to extrapolate the significance and implications of these findings within the broader context of Opinion Mining using Artificial Intelligence for digital media content analysis.

Our exploration encompasses a multifaceted examination of user sentiments, ranging from positive and negative to neutral, towards ChatGPT. We scrutinize the underlying reasons behind these sentiments, unveiling the driving forces behind user satisfaction, areas needing improvement, and the diverse array of discussions occurring within the digital realm.

Moreover, this discussion endeavors to shed light on the practical applications of these insights for quality assurance, decision-making, and future advancements within the landscape of AI-driven technologies. It aims to decipher how these findings can act as guiding principles for product enhancement, content curation, and strategic planning.

Ethical considerations and societal impacts stemming from AI-based Opinion Mining also find their place in this discussion. The ethical ramifications of user privacy, bias mitigation, and responsible AI deployment in digital media content analysis form an integral part of our discourse.

Additionally, limitations encountered during the study are critically examined, alongside recommendations aimed at refining methodologies and enhancing the robustness of AI-driven Opinion Mining approaches.

In essence, this chapter serves as a platform for a comprehensive and nuanced discussion, amalgamating findings, implications, and future prospects derived from the meticulous analysis conducted in the previous chapters. It aims to provide a deeper understanding of the significance and broader applications of AI-driven Opinion Mining in deciphering user sentiments and guiding decision-making within the realm of digital media content analysis.

5.2 Discussion

5.2.1 Analysis and Interpretation of Findings:

This section delves deep into the analysis of the findings obtained from sentiment analysis, topic modeling, and the identification of prevalent themes within positive, negative, and neutral tweets. It discusses the implications of these findings, highlighting key trends, patterns, and insights uncovered through AI-driven sentiment analysis techniques.

This segment entails a multifaceted examination of user sentiments, carefully dissecting and interpreting the outcomes obtained from the sentiment analysis process. It involves a meticulous scrutiny of tweets classified as positive, seeking to illuminate the underlying factors contributing to users' contentment, satisfaction, and positive perceptions of ChatGPT. This detailed analysis endeavors to unearth the specific aspects of ChatGPT that resonate positively with users, shedding light on successful user experiences, beneficial functionalities, or other aspects that drive favorable sentiments.

Simultaneously, this section rigorously dissects tweets categorized as negative sentiments, aiming to uncover areas of concern, limitations, or aspects of ChatGPT that provoke dissatisfaction among users. It delves into the intricacies of negative sentiments, discerning identified limitations, concerns, or perceived drawbacks within the ChatGPT framework, thereby contributing to a nuanced understanding of areas requiring improvement or refinement.

Furthermore, the examination extends to tweets classified as neutral sentiments, which neither exhibit overtly positive nor negative inclinations toward ChatGPT. This analysis aims to unveil the balanced perspectives, potential areas of user consensus, or broader discussions transcending polarized sentiments. It helps in deciphering discussions that may signify neutral attitudes, diverse opinions, or discussions of general interest surrounding ChatGPT without expressing explicit sentiments.

Overall, this exhaustive analysis aims not just to discern the polarities of sentiments but also to extrapolate deeper insights, decipher subtle shifts, and reveal nuanced aspects within sentiments. By scrutinizing the contexts, associations, and prevalent topics in tweets, this section provides comprehensive insights critical for informed decision-making, product enhancements, and future advancements in AI-driven technologies like ChatGPT.

5.2.2 Performance Comparison and Suitability of Methods

In this study, a comprehensive comparative analysis of several sentiment analysis methods, including VADER, linear regression, LSTM, naive Bayes, and random forest, was conducted to evaluate their performance in analyzing sentiments within digital media content. Each method exhibited distinctive strengths and weaknesses in several key aspects. In terms of accuracy, while LSTM demonstrated high accuracy, other methods like random forest and naive Bayes followed closely, whereas VADER and linear regression showed comparatively lower accuracy due to their reliance on specific rules or assumptions. Regarding

interpretability, linear regression and naive Bayes provided relatively clearer insights into sentiment analysis, whereas LSTM and random forest had complexities in interpreting their internal mechanisms. Adaptability-wise, LSTM showcased better adaptability to different types of digital content but demanded substantial computational resources, unlike VADER, which faced limitations due to predefined lexicons. Computational efficiency varied, with simpler methods like VADER and linear regression requiring less computation compared to LSTM and random forest, which were more resource-intensive.

Implementing these methods encountered challenges such as VADER's limitations in domain-specific contexts, linear regression's oversimplification of complex relationships, LSTM's resource-demanding nature for optimal performance, naive Bayes' independence assumption among features, and random forest's interpretability concerns and susceptibility to overfitting. Various use cases highlighted the proficiency of LSTM in handling diverse digital content due to its ability to capture context, whereas random forest and naive Bayes showcased effectiveness in specific domain-oriented sentiment analysis tasks. The implications of these methods in opinion mining from digital media content using AI are substantial, offering potential enhancements in decision-making, quality management, and user engagement across various industries relying on digital media content.

Proposed future directions include enhancing lexicon-based approaches like VADER, leveraging ensemble methods or deep learning architectures for sentiment analysis, and exploring hybrid models to capitalize on the strengths of individual methods. In conclusion, this comprehensive comparison and implementation of diverse sentiment analysis methods within the realm of digital media content using AI-driven approaches offer insights into their respective strengths, limitations, and applicability, paving the way for improved sentiment analysis techniques and further research exploration in this domain.

5.2.3 Understanding User Sentiments:

Discussion focuses on comprehending the sentiments expressed by users towards ChatGPT. It explores the reasons behind positive sentiments, such as user satisfaction, perceived benefits, and successful interactions, while dissecting the root causes of negative sentiments, including limitations, concerns, and areas needing improvement.

The focal point of this section revolves around deciphering the elements contributing to positive sentiments expressed by users. It delves into the reasons fostering a positive perception of ChatGPT, such as user satisfaction, perceived benefits, and successful interactions. By investigating these positive sentiments, the discussion endeavors to highlight the aspects of ChatGPT that resonate favorably with users. It may encompass the ease of interaction, the accuracy of responses, the diversity of functionalities, or the overall usefulness perceived by users. Identifying these elements provides invaluable insights into the strengths and advantageous aspects of ChatGPT that contribute to positive user experiences.

Simultaneously, this section rigorously dissects the root causes of negative sentiments expressed by users towards ChatGPT. It scrutinizes limitations, concerns, or areas necessitating improvement within the ChatGPT framework, aiming to unravel the reasons behind dissatisfaction or criticisms voiced by users. It may encompass identified limitations in functionality, perceived inefficiencies, complexities in interaction, or aspects that fail to meet user expectations. Uncovering these critical insights illuminates areas for improvement and refinement within ChatGPT, guiding strategies for enhancing user experiences and refining the model's functionalities.

By comprehensively exploring both positive and negative sentiments, this section seeks to provide a holistic understanding of user perspectives towards ChatGPT. It aims to identify the strengths and weaknesses perceived by users, thereby facilitating informed

decision-making, targeted improvements, and strategic enhancements in AI-driven technologies like ChatGPT. Ultimately, this understanding serves as a foundation for refining user experiences and guiding future developments to align with user expectations and preferences.

5.2.4 Implications for Quality Assurance and Decision-Making:

This section explores how the identified sentiments and topics can guide quality assurance measures, product enhancements, and strategic decisions. It elaborates on how insights derived from Opinion Mining can inform improvements in ChatGPT's functionalities, content curation, and user experiences.

Quality Assurance Measures: This section navigates through the implications drawn from sentiments and topics identified in ChatGPT-related discussions. It scrutinizes how these insights can be instrumental in shaping and bolstering quality assurance measures. By discerning the strengths and weaknesses highlighted in user sentiments, quality assurance protocols can be tailored to address identified concerns, refine functionalities, and ensure the model's efficacy and accuracy. For instance, identifying patterns of dissatisfaction or limitations in specific functionalities could prompt targeted testing and quality assurance efforts to rectify these issues, thereby enhancing ChatGPT's performance and reliability.

Product Enhancements: The discussion elaborates on how insights gleaned from Opinion Mining can steer product enhancement strategies for ChatGPT. By analyzing prevalent themes and sentiments, it identifies potential areas for improvement or innovation. For instance, if positive sentiments center around specific functionalities or user experiences, product enhancement efforts can be channeled towards augmenting those features. Similarly, addressing concerns highlighted in negative sentiments can drive targeted improvements in areas needing refinement, thereby elevating the overall quality of the ChatGPT model.

Strategic Decision-Making: This section outlines how insights derived from Opinion Mining can guide strategic decisions concerning ChatGPT. It delves into how the identified sentiments and prevailing topics can be leveraged to make informed decisions on future directions, investments, or innovations. Understanding user sentiments and preferences aids in aligning strategic goals with user expectations, thus enabling organizations to make proactive decisions regarding product development, marketing strategies, or resource allocation based on identified user needs and sentiments.

User Experience Refinement: Insights from sentiment analysis and topic identification can serve as a roadmap for refining user experiences within ChatGPT. By understanding what resonates positively with users and rectifying concerns voiced by them, efforts can be directed towards ensuring a seamless and satisfactory user journey. Enhancing user experiences based on these insights contributes to increased user satisfaction and engagement.

In essence, how insights derived from Opinion Mining hold substantial implications for elevating quality assurance, driving product enhancements, facilitating strategic decision-making, and refining user experiences within AI-driven technologies like ChatGPT. Leveraging these insights offers a comprehensive understanding of user sentiments, thereby paving the way for informed and effective decision-making processes.

5.2.5 Leveraging Insights for Future Development:

Discussion highlights the forward-looking aspect, utilizing insights from Opinion Mining to forecast future trends, technological advancements, and potential applications of ChatGPT. It explores how these insights can guide future research, innovations, and investments in AI-driven technologies. This forward-looking perspective aims to extrapolate from the insights gained through sentiment analysis and topic identification to forecast

potential trends, anticipate technological advancements, and identify prospective applications of ChatGPT in diverse domains.

Forecasting Future Trends: By analyzing the prevalent sentiments, emerging themes, and discussions surrounding ChatGPT, this section endeavors to forecast future trends in AI-driven technologies. Identifying patterns and trajectories in user sentiments can provide valuable indicators of evolving user preferences, expectations, and demands. These insights aid in predicting the direction of technological advancements and potential shifts in user behavior, thereby enabling organizations to anticipate and adapt to future trends in the AI landscape.

Guiding Future Research and Innovations: The discussion explores how insights derived from Opinion Mining can act as guiding beacons for future research endeavors and innovations in the realm of AI-driven technologies. By identifying areas of user interest, unmet needs, or potential enhancements highlighted in sentiments and topics, researchers and innovators can direct their efforts towards addressing these areas. This proactive approach aids in fostering innovation, pushing the boundaries of technology, and exploring novel avenues for the development of ChatGPT and similar AI models.

Informing Strategic Investments: Insights from Opinion Mining serve as a compass for strategic investments in AI-driven technologies. Organizations can use these insights to make informed decisions regarding resource allocation, R&D investments, and strategic partnerships. Understanding user sentiments and emerging themes guides investments towards areas of high user interest or areas needing improvement, ensuring that investments are aligned with user needs and market demands.

Identifying Potential Applications: The discussion also delves into exploring potential applications of ChatGPT based on the identified sentiments and prevalent topics. By understanding the contexts in which ChatGPT resonates positively or areas where

improvements are desired, stakeholders can identify diverse applications across industries.

These applications span from customer service and content generation to healthcare and education, thereby expanding the scope of AI-driven technologies and exploring new frontiers.

In summary, this section elucidates how leveraging insights from Opinion Mining can serve as a catalyst for driving future developments, innovations, and investments in AI-driven technologies like ChatGPT. By utilizing these insights proactively, organizations can steer their efforts towards aligning with evolving user needs, anticipating future trends, and fostering advancements that push the boundaries of AI technology.

5.2.6 Ethical and Societal Implications:

This segment delves into the ethical considerations arising from AI-based Opinion Mining and its impact on society. It discusses issues concerning user privacy, bias in sentiment analysis, and the responsible deployment of AI in digital media content analysis.

This segment aims to scrutinize various ethical concerns and societal impacts arising from the deployment of AI in analyzing user sentiments, guiding decisions, and shaping digital content.

User Privacy Concerns: This discussion delves into the ethical implications related to user privacy, emphasizing the importance of safeguarding user data while conducting Opinion Mining. It raises concerns about the collection, storage, and utilization of user-generated data in sentiment analysis processes. Analyzing user sentiments necessitates access to substantial user data, thus raising concerns about data privacy, consent, and ensuring compliance with privacy regulations. The discussion underscores the significance of ethical data handling practices and the imperative need to prioritize user privacy while conducting AI-driven content analysis.

Bias in Sentiment Analysis: Ethical considerations regarding bias in sentiment analysis algorithms are thoroughly explored in this segment. It delves into the challenges surrounding bias mitigation, ensuring fairness, and preventing algorithmic biases that may skew sentiment analysis results. Bias in AI models can lead to inaccurate analyses, perpetuate stereotypes, or result in unfair representations of certain demographics or opinions. Consequently, the discussion underscores the need for transparency, accountability, and continual monitoring to mitigate biases in AI-driven sentiment analysis algorithms.

Responsible AI Deployment: The ethical implications of responsibly deploying AI in digital media content analysis form a significant part of this discourse. It addresses the responsible use of AI in decision-making, content curation, and user interaction. It highlights the importance of ethical AI design, ensuring that AI models are deployed responsibly, transparently, and in a manner that upholds societal values, fairness, and human rights. Moreover, considerations regarding the impact of AI on societal discourse, public perceptions, and information dissemination are addressed, emphasizing the ethical responsibilities associated with deploying AI in digital media content analysis.

In summary, this section meticulously examines the ethical dimensions and societal implications arising from the utilization of AI-driven Opinion Mining in digital media content analysis. It aims to provoke thoughtful deliberation and encourages responsible practices, ethical considerations, and societal awareness in deploying AI for analyzing user sentiments, thereby ensuring that the benefits of AI are harnessed while mitigating potential ethical risks and societal impacts.

5.2.7 Limitations and Recommendations:

Acknowledging the limitations encountered during the study, this section discusses constraints in data collection, model accuracy, or interpretation of results. It offers

recommendations for refining methodologies, improving data collection, and enhancing the robustness of AI-driven Opinion Mining approaches.

Limitations Encountered:

The discussion begins by acknowledging and scrutinizing various limitations encountered throughout the study. These limitations might encompass constraints in data collection, potential biases in the dataset, inaccuracies in sentiment analysis algorithms, or challenges in the interpretation of results. For instance, limitations in data collection methods, such as the unavailability of certain data sources or the inability to access comprehensive datasets, could impact the comprehensiveness and representativeness of the analyses. Additionally, constraints related to model accuracy, including the presence of biases or inaccuracies in sentiment analysis algorithms, may have influenced the interpretations drawn from the data.

Insights from Previous Research:

Drawing insights from previous research studies in similar domains, this section correlates the encountered limitations with findings from existing literature. For instance, previous studies might have addressed challenges related to data biases, model inaccuracies, or interpretational issues in sentiment analysis within AI-driven content analysis. Referencing these studies helps provide context and further understanding of similar limitations and challenges faced in this research, establishing a continuum of knowledge and insights within the field.

Recommendations for Refinement:

Building upon the identified limitations, this segment offers strategic recommendations aimed at refining methodologies and improving the robustness of AI-driven Opinion Mining approaches. These recommendations could span diverse areas, such as refining data collection strategies, enhancing algorithmic accuracy, improving model

interpretability, or augmenting the overall reliability of sentiment analysis. For example, suggestions may include employing diverse data sources to mitigate biases, fine-tuning sentiment analysis models for greater accuracy, or incorporating advanced techniques for bias detection and mitigation.

Synthesis of Recommendations with Previous Research:

The recommendations put forth in this section are substantiated by insights garnered from prior research. This synthesis bridges the gap between encountered limitations, previous research findings, and the proposed recommendations. By aligning the recommendations with proven strategies and methodologies suggested in existing literature, this section provides a robust framework for addressing limitations and improving the methodologies employed in AI-driven Opinion Mining.

In essence, this section not only acknowledges the limitations faced but also offers valuable recommendations that draw from previous research findings. These recommendations serve as guiding principles for refining methodologies, mitigating challenges, and advancing the reliability and effectiveness of AI-driven Opinion Mining approaches within digital media content analysis.

5.3 Concluding Remarks:

The Discussion chapter culminates with a comprehensive summary, drawing together the main findings, implications, and future prospects highlighted throughout the study. It provides conclusive remarks, emphasizing the significance of AI-powered Opinion Mining in understanding user sentiments, guiding decision-making, and shaping the future of digital media content analysis.

This chapter serves as a critical platform to synthesize and critically analyze the study's outcomes, paving the way for a comprehensive understanding of the implications and

broader significance of the findings in the realm of AI-driven Opinion Mining for digital media content analysis.

Chapter VI.

Conclusion

6.1 Introduction

The final chapter of this thesis encapsulates the culminating insights and reflections derived from the comprehensive study undertaken to explore the application of Opinion Mining in analyzing digital media content, with a specific focus on ChatGPT Tweets Sentiment Analysis. This conclusive section consolidates the key takeaways, recommendations, and areas for future exploration distilled from the extensive examination of user sentiments, topic modeling, ethical considerations, and limitations encountered during the research.

The Introduction section sets the stage for the concluding remarks, providing an overview of the essential elements addressed in this chapter. It outlines the primary objectives of the thesis, briefly recaps the methodologies employed, and previews the subsequent sections devoted to conclusions, limitations, and future prospects. Within this framework, the chapter aims to present comprehensive closure by synthesizing the study's outcomes, implications, and avenues for further inquiry in the dynamic field of AI-driven Opinion Mining.

1.2 Conclusions and Recommendations

The implementation of a novel Opinion Mining Approach utilizing Artificial Intelligence to dissect sentiments within digital media content, particularly in the case of ChatGPT Sentiment Analysis, has brought forth an array of noteworthy findings and insights. This study sought to address several objectives, each contributing to a comprehensive understanding of sentiment analysis in social media platforms and the efficacy of AI-driven approaches in decision-making and quality assurance.

6.2.1 Summary of Achieved Objectives:

The research embarked on a journey to justify the significance of sentiment analysis in social media platforms, delve into existing literature, and propose an innovative approach for predicting sentiments from user reviews. Objective 1 illuminated the criticality of sentiment analysis in decoding user opinions, laying the foundation for improved decision-making processes. Objective 2 deepened our insights by exploring the diverse landscape of existing methodologies, paving the way for an inventive approach. Objective 3 culminated in a proposed framework involving sentiment extraction from text, audio, and video data, followed by the strategic clustering of sentiments into positive, negative, and neutral categories, ultimately enhancing the quality assurance and decision-making paradigms.

6.2.2 Addressing the Research Questions:

The research questions were meticulously addressed throughout the study. The importance of sentiment analysis was reaffirmed, highlighting the diverse applications it

holds across social media platforms, underscoring its indispensable role in understanding user sentiments. The study also scrutinized existing methods, showcasing the potential for enhancements and advancements in sentiment analysis techniques, promising a more nuanced and precise approach. Determining data collection sources and methodologies emerged as pivotal queries, guiding the research towards comprehensive data aggregation and extraction methodologies. Furthermore, the process of converting multi-modal user reviews, from video, audio, and image formats into text, was explored, presenting innovative possibilities for text-based sentiment analysis. The study successfully delved into methods for in-depth sentiment labeling and classification of tweets, enriching the analysis and providing nuanced insights into user sentiments.

6.2.3 Implications for Decision Making and Quality Assurance:

The findings of this research bear significant implications for decision-making processes and quality assurance within digital media content. The novel approach to Opinion Mining through AI offers a more refined understanding of user sentiments, thereby facilitating more informed decision-making processes. Additionally, the enhanced sentiment analysis techniques bring forth opportunities for quality assurance measures, ensuring improved content curation, and user satisfaction across various platforms.

The conclusive segment of this thesis synthesizes the key findings and insights drawn from the comprehensive exploration of Opinion Mining techniques applied to digital media content, specifically focusing on ChatGPT Tweets Sentiment Analysis. The conclusions encapsulate the core implications derived from sentiment analysis, topic modeling, and the ethical considerations highlighted in previous chapters. Based on these insights,

recommendations are proposed for enhancing AI-driven Opinion Mining methodologies, refining ChatGPT functionalities, and guiding future research and development.

6.2 Limitations of the Study

Acknowledging the boundaries and constraints encountered during this research, this section delineates the limitations that influenced the study's scope, data collection, methodology, and analyses. The discussion revolves around data limitations, potential biases, model constraints, and interpretational challenges that might have impacted the depth or comprehensiveness of the study's outcomes. Acknowledging these limitations is crucial for understanding the study's context and the boundaries of its implications.

It's crucial to acknowledge the limitations encountered during this research journey. Challenges pertaining to data collection, model accuracy, and interpretational nuances warrant further attention. Future studies could focus on refining data collection strategies, improving model accuracy, and exploring diverse sentiment analysis approaches to mitigate these limitations and augment the robustness of AI-driven Opinion Mining methodologies.

6.3 Future Work

This segment presents a forward-looking perspective, outlining potential avenues for future research, advancements, and applications in the realm of AI-driven Opinion Mining and digital media content analysis. It discusses unexplored areas, opportunities for methodological improvements, and potential enhancements in AI technologies like ChatGPT. These suggestions pave the way for further exploration, innovation, and advancements in leveraging Opinion Mining for insightful analysis of digital media content.

The Implementation of Novel Opinion Mining Approach on Digital Media Content using Artificial Intelligence has offered valuable insights and avenues for further exploration. This

study has expanded the horizons of sentiment analysis methodologies, emphasizing its pivotal role in decision-making and quality assurance in the digital landscape. As the field continues to evolve, the opportunities for leveraging AI in deciphering user sentiments and enhancing content analysis are both promising and boundless.

References

- Abdulrahman Radaideh1, F. D. (2023). Sentiment Analysis Predictions in Digital Media Content using NLP Techniques. *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 14, No. 11.
- Ahmad, M. S. (2017). Sentiment analysis of tweets using svm. *Int. J. Comput. Appl* 177, no. 5.
- al, M. A. (2020). Energy Choices in Alaska: Mining people's Perception and Attitudes from Geotagged Tweets,. *Renewable and Sustainable Energy Reviews*, vol. 124, p. 109781, May 2020.
- Albahri, A. S.-Q. (2020). Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (COVID-19): a systematic review. *Journal of medical systems* 44 (.
- Alomari, E. S. (2023). Malware detection using deep learning and correlation-based feature selection. *Symmetry* 15, no. 1 .
- Alotaibi, A. a. (2023). Enhancing the Sustainability of Deep-Learning-Based Network Intrusion Detection Classifiers against Adversarial Attacks. *Sustainability* 15, no. 12 .
- AlShahrani, B. M. (2021). Classification of cyber-attack using Adaboost regression classifier and securing the network. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 10.
- Al-Shareeda, M. A. (2023). DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison. *Bulletin of Electrical Engineering and Informatics* 12, no. 2.
- Alslaity, A. a. (2022). Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour & Information Technology* .
- Archak, N. A. (2007). Show me the money! Deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 56-65.
- Aslan, S. S. (2023). TSA-CNN-AOA: Twitter sentiment analysis using CNN optimized via arithmetic optimization algorithm. *Neural Computing and Applications*.

- Azam, Z. M. (2023). Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE*.
- Barry, J. (2017). Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches. In *AICS*, pp. 272-274.
- Bibi, M. W. (2022). A novel unsupervised ensemble framework using concept-based linguistic methods and machine learning for twitter sentiment analysis. *Pattern Recognition Letters* 158.
- Binali, H. V. (2009). A state of the art opinion mining and its application domains. In *2009 IEEE International Conference on Industrial Technology*, pp. 1-6. *IEEE*.
- Bock, F. E. (2019). A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials* 6.
- Boumans, J. W. (2018). Taking stock of the toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. *Rethinking Research Methods in an Age of Digital Journalism*.
- Braig, N. A. (2023). Machine Learning Techniques for Sentiment Analysis of COVID-19-Related Twitter Data. *IEEE Access* 11.
- Butt, U. A. (2023). Cloud-based email phishing attack using machine and deep learning algorithm. *Complex & Intelligent Systems* 9, no. 3 .
- Cambria, E. B. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems* 28, no. 2.
- Capatina, A. A.-E.-N. (2019). Knowledge maps for large-scale group decision making in social media content analysis. *Expert Systems: e13509*.
- Carvache-Franco, O. M.-F.-F. (2023). Topic and sentiment analysis of crisis communications about the COVID-19 pandemic in Twitter's tourism hashtags. *Tourism and Hospitality Research* 23, no. 1 .
- Castelo-Branco, F. J. (2020). Business intelligence and data mining to support sales in retail. In *Marketing and Smart Technologies: Proceedings of ICMarkTech 2019*, pp. 406-419. Springer Singapore.
- Catelli, R. S. (2023). Lexicon-based sentiment analysis to detect opinions and attitude towards COVID-19 vaccines on Twitter in Italy. *Computers in Biology and Medicine* 158.
- Chee, R. M. (2023). The impact of social media influencers on pregnancy, birth, and early parenting experiences: A systematic review. *Midwifery*.
- Dogan, A. a. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications* 166.

- E. Cambria, Y. L. (2020). SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, vol. 28,.
- Elmitwally, A. a. (2020). A Comprehensive Study for Arabic Sentiment Analysis (Challenges and Applications),. *Egyptian Informatics Journal*, vol.21,no.1,pp.7–12,Mar.2020,doi: <https://doi.org/10.1016/j.eij.2019.06.001>.
- Fanni, S. C. (2023). Natural language processing. In *Introduction to Artificial Intelligence*, pp. 87-99. Cham: Springer International Publishing.
- Ghermandi, A. a. (2019). Passive crowdsourcing of social media in environmental research: A systematic map. *Global environmental change* 55.
- Guo, Y. W. (2023). Intelligent manufacturing management system based on data mining in artificial intelligence energy-saving resources. *Soft Computing* 27, no. 7 .
- Hackman, J. R. (1995). Total quality management: Empirical, conceptual, and practical issues. *Administrative science quarterly* .
- Hamad, E. O. (2016). Toward a mixed-methods research approach to content analysis in the digital age: the combined content-analysis model and its applications to health care Twitter feeds. *Journal of medical Internet research* 18, no. 3 (.
- Hasan, M. R. (2019). Sentiment analysis with NLP on Twitter data. In *2019 international conference on computer, communication, chemical, materials and electronic engineering (IC4ME2)*, pp. 1-4.
- Iio, J. (2023). Analysis of Critical Comments on ChatGPT. In *International Conference on Network-Based Information Systems*, pp. 455-463. Cham: Springer Nature Switzerland.
- Imran, M. H. (2023). A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems. *Computers & Security* 134.
- Iparraguirre-Villanueva, O. A.-R.-C.-P.-C. (2023). The public health contribution of sentiment analysis of Monkeypox tweets to detect polarities using the CNN-LSTM model. *Vaccines* 11, no. 2.
- Joloudari, J. H. (2019). "BERT-deep CNN: State of the art for sentiment analysis of COVID-19 tweets. *Social Network Analysis and Mining* 13, no. 1.
- Kabiri, M. E. (2019). HOMPer: A New Hybrid System for Opinion Mining in the Persian Language. *Journal of Information Science*, vol. 46, no. 1, pp. 101–117, .
- Kaur, G. a. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of Big Data* 10, no. 1.
- Korkmaz, A. C. (2023). Analyzing the User's Sentiments of ChatGPT Using Twitter Data. *Iraqi Journal For Computer Science and Mathematics* 4, no.

- KORKMAZ, A. C. (2023). Sentiment Analysis of ChatGPT Using Twitter Data.
- Küçük, D. a. (2019). Deep Learning-Based Sentiment and Stance Analysis of Tweets About Vaccination. *International Journal on Semantic Web and Information Systems (IJSWIS) 19*.
- Kumar, M. R. (2021). Data mining and machine learning in retail business: developing efficiencies for better customer retention. *Journal of Ambient Intelligence and Humanized Computing*.
- Kumari, R. a. (2023). Twitter Sentiment Analysis using Machine Learning Techniques: A Case Study of ChatGPT. In *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, vol. 1, pp. 1-5. IEEE.
- Leelawat, N. S. (2022). Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning. *Heliyon 8*.
- Li, J. A. (2020). Blockchain for supply chain quality management: challenges and opportunities in context of open manufacturing and industrial internet of things. *International Journal of Computer Integrated Manufacturing 33, no. 12* .
- Li, R. Y.-Q.-Q.-N.-Y. (2023). Sentiment mining of online reviews of peer-to-peer accommodations: Customer emotional heterogeneity and its influencing factors. *Tourism Management 96* .
- Ling, C. X. (1998). Data mining for direct marketing: Problems and solutions. In *Kdd*, vol. 98, pp. 73-79.
- M. Anwer, S. M. (2021). Attack Detection in IoT using Machine Learning. *Eng. Technol. Appl. Sci. Res., vol. 11, no. 3, pp. 7273–7278*.
- M. D. Nguyen, P. Q. (2023). An Application of Analytic Network Process (ANP) to Assess Critical Risks of Bridge Projects in the Mekong Delta Region". *Eng. Technol. Appl. Sci. Res., vol. 13, no. 3, pp. 10622–10629*.
- Madhu, H. S. (2023). Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. *Expert Systems with Applications 215*.
- Mann, S. J. (2022). Twitter sentiment analysis using enhanced BERT. In *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*,.
- Mardjo, A. a. (2022). HyVADRF: Hybrid VADER–Random Forest and GWO for Bitcoin Tweet Sentiment Analysis. *IEEE Access 10``*.
- Martínez-Plumed, F. L.-O.-O.-Q. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering 33, no. 8* .
- Mathur, M. K. (2023). Digital Marketing and Its Effect on Startups. *European Economic Letters (EEL) 13, no. 1s*.

- Messaoudi, C. Z. (2022). Opinion mining in online social media: a survey. *Social Network Analysis and Mining* 12, no. 1.
- Msugther, A. E. (2023). Artificial Intelligence and the Media: Revisiting Digital Dichotomy Theory. In *Information Systems Management*. IntechOpen.
- Mujahid, M. F. (2023). Analyzing Sentiments Regarding ChatGPT Using Novel BERT: A Machine Learning Approach. *Information* 14, no. 9.
- Musleh, D. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *Journal of Sensor and Actuator Networks* 12, no. 2.
- Naderi, H. a. (2023). Digital twinning of civil infrastructures: Current state of model architectures, interoperability solutions, and future prospects. *Automation in Construction* 149.
- Neethu, M. S. (2013). Sentiment analysis in twitter using machine learning techniques. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*.
- Ning He, Q. W. (n.d.). Applied Energy, 2024, Volume 353, Page 122048. 2023.
- O. Araque, I. C.-P.-R. (2017). Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications. *Expert Systems with Applications*, vol. 77, .
- Pang, B. a. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval* 2, no. 1–2 .
- Parvin, S. A. (2023). A Novel Approach to Classify Sentiments on Different Datasets Using Hybrid Approaches of Sentiment Analysis. *Indian Journal of Science and Technology* 16, no. 44 .
- Polat, H. O. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* 12, no. 3 .
- Ramasangu, S. R. (2023). Classification of Cognitive States using Task-Specific Connectivity Features. *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 3, pp. 10675–10679.
- Rastogi, R. a. (2023). Diabetes prediction model using data mining techniques. *Measurement: Sensors* 25.
- Ravi, K. a. (2015). "A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-based systems* 89 .
- Ravi, K. a. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-based systems* 89 .
- Rigaki, M. a. (2023). A survey of privacy attacks in machine learning. *ACM Computing Surveys* 56, no. 4.
- Rudkowsky, E. M. (2018). More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures* 12, no. 2-3.

- Sarker, I. H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. *Annals of Data Science* 10, no. 6.
- Shamrat, F. M. (2021). Sentiment analysis on twitter tweets about COVID-19 vaccines using NLP and supervised KNN classification algorithm. *Indonesian Journal of Electrical Engineering and Computer Science* 23, no. 1 .
- Sharma, S. R. (2023). "Mining Twitter for Insights into ChatGPT Sentiment: A Machine Learning Approach. In 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), pp. 1-6. IEEE,.
- Shrivastav, S. K. (2023). Exploring the application of analytics in supply chain during COVID-19 pandemic: a review and future research agenda. *Journal of Global Operations and Strategic Sourcing* 16, no. 2.
- Srinita, S. a. (2023). Investigating the resilience of micro, small and medium enterprises in entering the digital market us-ing social media: Evidence from Aceh province, Indonesia. *International Journal of Data and Network Science* 7, no. 4.
- Stieglitz, S. M. (2018). The Adoption of social media analytics for crisis management– Challenges and Opportunities.
- Su, Y. a. (2022). Public Perception of ChatGPT and Transfer Learning for Tweets Sentiment Analysis Using Wolfram Mathematica. *Data* 8, no. 12 .
- Sufi, F. (2022). Algorithms in low-code-no-code for research applications: a practical review. *Algorithms* 16, no. 2.
- Tsui, K.-L. V. (2023). "Data mining methods and applications. In *Springer handbook of engineering statistics*, pp. 797-816. London: Springer London.
- Wang, Z. Q. (2023). Is ChatGPT a good sentiment analyzer? A preliminary study. *arXiv preprint arXiv:2304.04339*.
- Wang, Z. X. (2023). "Intrusion detection and network information security based on deep learning algorithm in urban rail transit management system. *IEEE Transactions on Intelligent Transportation Systems* 24, no. 2.
- Waqas, M. S. (2022). The role of artificial intelligence and machine learning in wireless networks security: Principle, practice and challenges. *Artificial Intelligence Review* 55, no. 7.
- Xu, A. M. (2023). Sentiment Analysis On Twitter Posts About The Russia and Ukraine War With Long Short-Term Memory. *Sinkron: jurnal dan penelitian teknik informatika* 8, no. 2 .
- Xu, H. Z. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing* 27, no. 19 (2023): 14469-14481.

- Yadav, V. P. (2023). Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *International Journal of Information Technology* 15, no. 2 .
- Zonnenshain, A. a. (2020). Quality 4.0—the challenging future of quality engineering. *Quality Engineering* 32, no. 4.
- Bello, A., Ng, S. C., & Leung, M. F. (2023). A BERT framework to sentiment analysis of tweets. *Sensors*, 23(1), 506.
- Bello, A., Ng, S. C., & Leung, M. F. (2023). A BERT framework to sentiment analysis of tweets. *Sensors*, 23(1), 506.
- Zhao, G., Luo, Y., Chen, Q., & Qian, X. (2023). Aspect-based sentiment analysis via multitask learning for online reviews. *Knowledge-Based Systems*, 264, 110326.
- Liu, B. (2022). *Sentiment analysis and opinion mining*. Springer Nature.
- Shaik, T., Tao, X., Dann, C., Xie, H., Li, Y., & Galligan, L. (2023). Sentiment analysis and opinion mining on educational data: A survey. *Natural Language Processing Journal*, 2, 100003.
- Kumar, R. S., Saviour Devaraj, A. F., Rajeswari, M., Julie, E. G., Robinson, Y. H., & Shanmuganathan, V. (2021). Exploration of sentiment analysis and legitimate artistry for opinion mining. *Multimedia Tools and Applications*, 1-16.
- Păvăloaia, V. D., Teodor, E. M., Fotache, D., & Danileț, M. (2019). Opinion mining on social media data: sentiment analysis of user preferences. *Sustainability*, 11(16), 4459.
- Riaz, S., Fatima, M., Kamran, M., & Nisar, M. W. (2019). Opinion mining on large scale data using sentiment analysis and k-means clustering. *Cluster Computing*, 22, 7149-7164.
- Elbagir, S., & Yang, J. (2019, March). Twitter sentiment analysis using natural language toolkit and VADER sentiment. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 122, p. 16).
- Tymann, K., Lutz, M., Palsbröker, P., & Gips, C. (2019, September). GerVADER-A German Adaptation of the VADER Sentiment Analysis Tool for Social Media Texts. In *LWDA* (pp. 178-189).
- Bose, D. R., Aithal, P. S., & Roy, S. (2021). Survey of twitter viewpoint on application of drugs by VADER sentiment analysis among distinct countries. *International Journal of Management, Technology, and Social Sciences (IJMITS)*, 6(1), 110-127.
- Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. (2021). LSTM, VADER and TF-IDF based hybrid sentiment analysis model. *International Journal of Advanced Computer Science and Applications*, 12(7).
- Biswas, S., & Ghosh, S. (2022). Drug Usage Analysis by VADER Sentiment Analysis on Leading Countries. *Mapana Journal of Sciences*, 21(3).

- al, M. A. (2020). Energy Choices in Alaska: Mining people's Perception and Attitudes from Geotagged Tweets,. *Renewable and Sustainable Energy Reviews*, vol. 124, p. 109781, May 2020.
- al., G. A. (2019). AI in the media and creative industries. *New European Media* , vol. 01.
- Alslaity, A. a. (2022). Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour & Information Technology* .
- Bibi, M. W. (2022). A novel unsupervised ensemble framework using concept-based linguistic methods and machine learning for twitter sentiment analysis. *Pattern Recognition Letters* 158.
- Braig, N. A. (2023). Machine Learning Techniques for Sentiment Analysis of COVID-19- Related Twitter Data. *IEEE Access* 11.
- Carvache-Franco, O. M.-F.-F. (2023). Topic and sentiment analysis of crisis communications about the COVID-19 pandemic in Twitter's tourism hashtags. *Tourism and Hospitality Research* 23, no. 1 .
- Catelli, R. S. (2023). Lexicon-based sentiment analysis to detect opinions and attitude towards COVID-19 vaccines on Twitter in Italy. *Computers in Biology and Medicine* 158.
- E. Cambria, B. S. (2018). New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, vol. 28, no. 2.
- E. Cambria, Y. L. (2020). SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, vol. 28,.
- Elmitwally, A. a. (2020). A Comprehensive Study for Arabic Sentiment Analysis (Challenges and Applications),. *Egyptian Informatics Journal*, vol.21,no.1,pp.7–12,Mar.2020,doi: <https://doi.org/10.1016/j.eij.2019.06.001>.
- Fanni, S. C. (2023). Natural language processing. In *Introduction to Artificial Intelligence*, pp. 87-99. Cham: Springer International Publishing.
- Iparraguirre-Villanueva, O. A.-R.-C.-P.-C. (2023). The public health contribution of sentiment analysis of Monkeypox tweets to detect polarities using the CNN-LSTM model. *Vaccines* 11, no. 2.
- Kabiri, M. E. (2019). HOMPer: A New Hybrid System for Opinion Mining in the Persian Language. *Journal of Information Science*, vol. 46, no. 1, pp. 101–117, .
- Kaur, G. a. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of Big Data* 10, no. 1.
- Leelawat, N. S. (2022). Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning. *Heliyon* 8.

- M. Anwer, S. M. (2021). Attack Detection in IoT using Machine Learning. *Eng. Technol. Appl. Sci. Res.*, vol. 11, no. 3, pp. 7273–7278.
- M. D. Nguyen, P. Q. (2023). An Application of Analytic Network Process (ANP) to Assess Critical Risks of Bridge Projects in the Mekong Delta Region”. *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 3, pp. 10622–10629.
- Madhu, H. S. (2023). Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. *Expert Systems with Applications* 215.
- Mann, S. J. (2022). Twitter sentiment analysis using enhanced BERT. In *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*,.
- Muhammad, S. H. (2023). Afrisenti: A twitter sentiment analysis benchmark for african languages. *arXiv preprint arXiv:2302.08956*.
- Neethu, M. S. (2013). Sentiment analysis in twitter using machine learning techniques. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*.
- O. Araque, I. C.-P.-R. (2017). Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications. *Expert Systems with Applications*, vol. 77,.
- Park, S. S. (2023). Mind games: A temporal sentiment analysis of the political messages of the Internet Research Agency on Facebook and Twitter. *New Media & Society* 25, no. 3 .
- Ramasangu, S. R. (2023). Classification of Cognitive States using Task-Specific Connectivity Features. *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 3, pp. 10675–10679.
- Shrivastav, S. K. (2023). Exploring the application of analytics in supply chain during COVID-19 pandemic: a review and future research agenda. *Journal of Global Operations and Strategic Sourcing* 16, no. 2.
- Sufi, F. (2022). Algorithms in low-code-no-code for research applications: a practical review. *Algorithms* 16, no. 2.
- U. A. Chauhan, M. T. (2020). A Comprehensive Analysis of Adverb Types for Mining User Sentiments on Amazon Product Reviews. *Wide Web*, vol. 23, no. 3, pp. 1811–1829.
- Wang, N. C. (2018). Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis. In *Proceedings of 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, IEEE.
- Xu, A. M. (2023). Sentiment Analysis On Twitter Posts About The Russia and Ukraine War With Long Short-Term Memory. *Sinkron: jurnal dan penelitian teknik informatika* 8, no. 2 .
- Yadav, V. P. (2023). Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *International Journal of Information Technology* 15, no. 2 .

- al, M. A. (2020). Energy Choices in Alaska: Mining people's Perception and Attitudes from Geotagged Tweets,. *Renewable and Sustainable Energy Reviews*, vol. 124, p. 109781, May 2020.
- al., G. A. (2019). AI in the media and creative industries. *New European Media* , vol. 01.
- Alslaity, A. a. (2022). Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour & Information Technology* .
- Bibi, M. W. (2022). A novel unsupervised ensemble framework using concept-based linguistic methods and machine learning for twitter sentiment analysis. *Pattern Recognition Letters* 158.
- Braig, N. A. (2023). Machine Learning Techniques for Sentiment Analysis of COVID-19- Related Twitter Data. *IEEE Access* 11.
- Carvache-Franco, O. M.-F.-F. (2023). Topic and sentiment analysis of crisis communications about the COVID-19 pandemic in Twitter's tourism hashtags. *Tourism and Hospitality Research* 23, no. 1 .
- Catelli, R. S. (2023). Lexicon-based sentiment analysis to detect opinions and attitude towards COVID-19 vaccines on Twitter in Italy. *Computers in Biology and Medicine* 158.
- E. Cambria, B. S. (2018). New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, vol. 28, no. 2.
- E. Cambria, Y. L. (2020). SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, vol. 28,.
- Elmitwally, A. a. (2020). A Comprehensive Study for Arabic Sentiment Analysis (Challenges and Applications),. *Egyptian Informatics Journal*, vol.21,no.1,pp.7–12, Mar.2020,doi: <https://doi.org/10.1016/j.eij.2019.06.001>.
- Fanni, S. C. (2023). Natural language processing. In *Introduction to Artificial Intelligence*, pp. 87-99. Cham: Springer International Publishing.
- Iparraguirre-Villanueva, O. A.-R.-C.-P.-C. (2023). The public health contribution of sentiment analysis of Monkeypox tweets to detect polarities using the CNN-LSTM model. *Vaccines* 11, no. 2.
- Kabiri, M. E. (2019). HOMPer: A New Hybrid System for Opinion Mining in the Persian Language. *Journal of Information Science*, vol. 46, no. 1, pp. 101–117, .
- Kaur, G. a. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of Big Data* 10, no. 1.

- Leelawat, N. S. (2022). Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning. *Heliyon* 8.
- M. Anwer, S. M. (2021). Attack Detection in IoT using Machine Learning. *Eng. Technol. Appl. Sci. Res.*, vol. 11, no. 3, pp. 7273–7278.
- M. D. Nguyen, P. Q. (2023). An Application of Analytic Network Process (ANP) to Assess Critical Risks of Bridge Projects in the Mekong Delta Region". *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 3, pp. 10622–10629.
- Madhu, H. S. (2023). Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. *Expert Systems with Applications* 215.
- Mann, S. J. (2022). Twitter sentiment analysis using enhanced BERT. In *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*.
- Muhammad, S. H. (2023). Afrisenti: A twitter sentiment analysis benchmark for african languages. *arXiv preprint arXiv:2302.08956*.
- Neethu, M. S. (2013). Sentiment analysis in twitter using machine learning techniques. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*.
- O. Araque, I. C.-P.-R. (2017). Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications. *Expert Systems with Applications*, vol. 77,.
- Park, S. S. (2023). Mind games: A temporal sentiment analysis of the political messages of the Internet Research Agency on Facebook and Twitter. *New Media & Society* 25, no. 3 .
- Ramasangu, S. R. (2023). Classification of Cognitive States using Task-Specific Connectivity Features. *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 3, pp. 10675–10679.
- Shrivastav, S. K. (2023). Exploring the application of analytics in supply chain during COVID-19 pandemic: a review and future research agenda. *Journal of Global Operations and Strategic Sourcing* 16, no. 2.
- Sufi, F. (2022). Algorithms in low-code-no-code for research applications: a practical review. *Algorithms* 16, no. 2.
- U. A. Chauhan, M. T. (2020). A Comprehensive Analysis of Adverb Types for Mining User Sentiments on Amazon Product Reviews. *Wide Web*, vol. 23, no. 3, pp. 1811–1829.
- Wang, N. C. (2018). Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis. In *Proceedings of 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, IEEE.
- Xu, A. M. (2023). Sentiment Analysis On Twitter Posts About The Russia and Ukraine War With Long Short-Term Memory. *Sinkron: jurnal dan penelitian teknik informatika* 8, no. 2 .

Yadav, V. P. (2023). Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *International Journal of Information Technology* 15, no. 2 .

APPENDICES

Appendix A Novel method for Detection of ChatGPT

```
data = pd.read_csv("digitalmedia.csv")  
  
data = data.drop([data.columns[0]], axis=1)  
  
data.values[:5].tolist()
```

[2]:

```
[['ChatGPT: Optimizing Language Models for Dialogue https://t.co/K9rKRygYyn  
@OpenAI',  
'neutral'],  
 ['Try talking with ChatGPT, our new AI system which is optimized for dialogue. Your  
 feedback will help us improve it. https://t.co/sHDm57g3Kr',  
'good'],  
 ['ChatGPT: Optimizing Language Models for Dialogue https://t.co/GLEbMoKN6w #AI  
 #MachineLearning #DataScience #ArtificialIntelligence\\n\\nTrending AI/ML Article  
 Identified & Digest via Granola; a Machine-Driven RSS Bot by Ramsey Elbasheer  
https://t.co/RprmAXUp34',  
'neutral'],  
 ['THRILLED to share that ChatGPT, our new model optimized for dialog, is now public,  
 free, and accessible to everyone. https://t.co/dyvtHecYbd https://t.co/DdhzhqhCBX  
https://t.co/l8qTLure71',  
'good'],
```

['As of 2 minutes ago, @OpenAI released their new ChatGPT. \n\nAnd you can use it right now ↗ <https://t.co/VyPGPNw988> <https://t.co/cSn5h6h1M1>',
'bad']]

data.head()

Word Cloud of data, before cleaning

```
from wordcloud import WordCloud
```

```
text_data = data['tweets'].str.cat(sep=' ')
```

```
# Create a WordCloud object
```

```
wordcloud = WordCloud(width=800, height=400,  
background_color="#F5F5F5").generate(text_data)
```

```
# Display the word cloud using Matplotlib
```

```
plt.figure(figsize=(10, 5))  
plt.imshow(wordcloud, interpolation='bilinear')  
plt.axis('off')  
plt.show()
```

add Codeadd Markdown

Drop duplicated values

```
print("Duplicated values: " , data.duplicated().sum())
```

```
data.drop_duplicates(inplace=True)
```

```
data = data.dropna(axis=0)
```

```
print(data.info())
Duplicated values: 1671
<class 'pandas.core.frame.DataFrame'>
Int64Index: 217623 entries, 0 to 219293
Data columns (total 2 columns):
 #  Column Non-Null Count Dtype 
 --- 
 0  tweets  217623 non-null object 
 1  labels   217623 non-null object 
dtypes: object(2)
memory usage: 5.0+ MB
```

None

add Codeadd Markdown

balance data

cleaning dataset

import nltk

```
from nltk.corpus import wordnet as wn
from nltk.stem import WordNetLemmatizer,PorterStemmer
```

```
nltk.download('wordnet')
```

```
nltk.download('punkt')
```

```
nltk.download('stopwords')
```

```
stop_words = set(stopwords.words('english'))
```

```
st = PorterStemmer()
```

```

lem = WordNetLemmatizer()

def is_alpha(word):
    for part in word.split('-'):
        if not part.isalpha():
            return False

    return True

def clean_dataset(text):
    text = re.sub(r'http\S+', "", text) # removing links

    text = re.sub(r'\n', ' ', text) # removing \n

    text = re.sub(r"\s*#\S+", "", text) # removing hash tags

    text = re.sub(r"\s*@\S+", "", text) # removing @

    text = text.lower()

    words = [word for word in word_tokenize(text) if is_alpha(word)]
    #words = [st.stem(word) for word in words]

    words = [lem.lemmatize(word) for word in words]

    # text = " ".join([word for word in text.split(" ") if is_alpha(word)])
    # text = re.sub(r'^[a-zA-Z\s]', ' ', text, re.I/re.A)

    words = [w for w in words if w not in stop_words]

    text = " ".join(words)

return text.strip()

```

Word Cloud of data, after cleaning

```
from wordcloud import WordCloud

text_data = data['cleaned_tweets'].str.cat(sep=' ')

# Create a WordCloud object
wordcloud = WordCloud(width=800, height=400,
background_color="#F5F5F5").generate(text_data)
```

Display the word cloud using Matplotlib

```
plt.figure(figsize=(10, 5))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.show()
```

add Codeadd Markdown

Converting dataset in numerical form

**

```
import math

import collections
```

```
def convert_text_to_numerical(text):

    num_words = 7000

    tokenizer = Tokenizer(num_words=num_words)
```

```

tokenizer.fit_on_texts(text)

sequences = tokenizer.texts_to_sequences(text)

#maxlen = max(45, math.ceil(np.average([len(seq) for seq in sequences])))
maxlen = 140

pad_seqs = pad_sequences(sequences, maxlen=maxlen)

pad_seqs_todrop = []

for i, p in enumerate(pad_seqs):

    if sum(p) == sum(sorted(p, reverse=True)[0:2]):

        pad_seqs_todrop.append(i)

return pad_seqs, pad_seqs_todrop, tokenizer, num_words, maxlen

```

```

data = data.reset_index()

numeric_tweets, rows_todrop, tokenizer, num_words, maxlen =

convert_text_to_numerical(data['cleaned_tweets'])

data.insert(len(data.columns)-1, "numeric_tweets", numeric_tweets.tolist())

data.head()

```

Splitting Data: training 80% and 20% for testing

*#@title ***Splitting the dataset into training and testing sets****

```

inputs = final_data[['tweets', 'cleaned_tweets', 'numeric_tweets']]

outputs = final_data[['labels', 'encoded_labels']]

```

```
in_train, in_test, out_train, out_test = train_test_split(inputs, outputs, test_size=0.2,  
shuffle=True, random_state=42)
```

```
X_train = in_train['numeric_tweets']
```

```
X_test = in_test['numeric_tweets']
```

```
y_train = out_train['encoded_labels']
```

```
y_test = out_test['encoded_labels']
```

add Codeadd Markdown

Saving training and testing data into csv files

```
#training_df = pd.concat([in_train, out_train], axis=1)  
  
#training_df.to_csv('training_data.csv', encoding='utf-8', index=False)  
  
#testing_df = pd.concat([in_test, out_test], axis=1)  
  
#testing_df.to_csv('testing_data.csv', encoding='utf-8', index=False)  
  
X_train = X_train.astype(np.int32)  
  
X_test = np.asarray(X_test.tolist()).astype(np.int32)  
  
y_train = np.asarray(y_train.tolist()).astype(np.int32)  
  
y_test = np.asarray(y_test.tolist()).astype(np.int32)
```

```
type(X_train)
```

[17]:

```
numpy.ndarray
```

Lstm

```
from keras.preprocessing.text import Tokenizer  
  
from keras.utils import pad_sequences
```

```
# Tokenize the input text
```

```

tokenizer = Tokenizer(num_words=7000)

tokenizer.fit_on_texts(data["tweets"])

# Pad the sequences to a fixed length

max_len = 140

add Codeadd Markdown

word embeding

import numpy as np

from keras.utils import to_categorical

# Load pre-trained word embeddings

embedding_dim = 100

embeddings_index = {}

with open('/kaggle/input/glove6b100dtxt/glove.6B.100d.txt', encoding='utf8') as f:

    for line in f:

        values = line.split()

        word = values[0]

        coefs = np.asarray(values[1:], dtype='float32')

        embeddings_index[word] = coefs

# Create an embedding matrix for the tokenizer

word_index = tokenizer.word_index

#num_words = min(len(word_index), 7000)

num_words = 7000

```

```
embedding_matrix = np.zeros((num_words, embedding_dim))

for word, i in word_index.items():

    if i >= num_words:

        continue

    embedding_vector = embeddings_index.get(word)

    if embedding_vector is not None:

        embedding_matrix[i] = embedding_vector
```

Convert the output labels to one-hot encoded vectors

```
y_train_en = to_categorical(y_train)
```

```
y_test_en = to_categorical(y_test)
```

add Codeadd Markdown

training model

```
from keras.models import Sequential

from keras.layers import Embedding, LSTM, Dense

from keras.callbacks import EarlyStopping
```

Create the model

```
model2 = Sequential()

model2.add(Embedding(num_words, embedding_dim, input_length=max_len,
weights=[embedding_matrix], trainable=True))

model2.add(LSTM(64, dropout=0.3, recurrent_dropout=0.3))

model2.add(Dense(3, activation='softmax'))

#model2.add(Dense(3, activation='relu'))
```

```

# Compile the model

model2.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model

#model2.fit(X_train, y_train_onehot, validation_data=(X_test, y_test_onehot), epochs=10,
batch_size=1024)

history2 = model2.fit(X_train, y_train_en, epochs=10, batch_size=1024, validation_split=0.1,
callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.001)])

def get_measurements(true_y, pred_y, average='micro'):

return {

    "accuracy": accuracy_score(true_y, pred_y),
    "recall": recall_score(true_y, pred_y, average=average),
    "precision": precision_score(true_y, pred_y, average=average),
    "fscore": f1_score(true_y, pred_y, average=average),
}

predicted_labels = model2.predict(X_test, verbose=1)

{'accuracy': 0.8919144621074294,
'recall': 0.8919144621074294,
'precision': 0.8919144621074294,
'fscore': 0.8919144621074294}

def predict_user_input(tweets , model):

    data = pd.DataFrame({ 'Tweets': tweets })

    cleaned_data = data['Tweets'].apply(clean_dataset)

```

```

sequences = tokenizer.texts_to_sequences(cleaned_data)

pad_seqs = pad_sequences(sequences, maxlen=maxlen)

predicted_labels = model.predict(pad_seqs)

data['labels'] = [ label_encoder.classes_[label_num] for label_num in

np.argmax(predicted_labels, axis=1)]

#print(data)

return predicted_labels, data


print(predict_user_input([
    'chatgpt is very crazy !',
    'chatgpt is kind a dangerous',
    'woooow, chatgt is really impressive !!!',
    "chatgpt is an AI tool",
    "good results"] , model2))

print("")

#x = [1 , 2 , 3 , 4]

x= np.array([1, 2 , 3 , 4])

y = [52 , 86 , 75 , 88]

y= np.array([52 , 86 , 75 , 88])

#plt.plot(x)

plt.plot(x,y)

plt.xlabel("Trial")

```

```
plt.ylabel("Accuracy")
```

```
plt.show()
```

```
x = [1 , 2 , 3 , 4]
```

```
y = [40 , 86 , 89, 89]
```

```
#plt.plot(x)
```

```
plt.plot(x,y)
```

```
plt.xlabel("Trial")
```

```
plt.ylabel("Accuracy")
```

```
plt.show()
```

```
plot_graphs(history2, "accuracy")
```

```
plot_graphs(history2, "loss")
```

```
def userIn(model):
```

```
    x=input("enter sentence or x for exit: ")
```

```
    while x != "x":
```

```
        predict_user_input([x] , model)
```

```
        x=input("enter sentence or x for exit: ")
```

```
userIn(model2)
```

LOGISTIC REGRESSION

add Codeadd Markdown

Scaling data to train

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

add Codeadd Markdown

Model training

```
from sklearn.linear_model import LogisticRegression
```

```
log=LogisticRegression()
```

```
log.fit(X_train_scaled,y_train)
```

add Codeadd Markdown

Prediction

```
log_pred = log.predict(X_test_scaled)
```

add Codeadd Markdown

Calculating Metrics

Calculate accuracy

```
accuracy = accuracy_score(y_test, log_pred)
```

Calculate precision

```
precision = precision_score(y_test, log_pred, average='weighted')
```

Calculate recall

```
recall = recall_score(y_test, log_pred, average='weighted')
```

```

print("Accuracy : ",accuracy)

print("Precision : ",precision)

print('Recall : ',recall)

import numpy as np

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_curve, auc

from sklearn.preprocessing import label_binarize

# Convert y_test to binary labels

y_test_binary = label_binarize(y_test, classes=np.unique(y_test))

# Calculate the scores for each class

y_pred_scores = log.predict_proba(X_test_scaled)

# Compute the false positive rate (fpr), true positive rate (tpr), and area under the ROC curve
(auc) for each class

fpr = dict()

tpr = dict()

roc_auc = dict()

n_classes = y_test_binary.shape[1]

for i in range(n_classes):

    fpr[i], tpr[i], _ = roc_curve(y_test_binary[:, i], y_pred_scores[:, i])

    roc_auc[i] = auc(fpr[i], tpr[i])

```

```

# Plot ROC curve for each class

plt.figure()

for i in range(n_classes):

    plt.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f)' % roc_auc[i])



plt.plot([0, 1], [0, 1], 'k--') # Diagonal line

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve')

plt.legend(loc="lower right")

plt.show()

```

add Codeadd Markdown

Naive Bayes

add Codeadd Markdown

Scaling data using min-max

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)
```

```
from sklearn.naive_bayes import MultinomialNB
```

```

nb_classifier = MultinomialNB()

# Train the classifier

nb_classifier.fit(X_train_scaled, y_train)

# Predict on the test set

y_pred = nb_classifier.predict(X_test)

# Calculate accuracy

accuracy = accuracy_score(y_test, y_pred)

# Calculate precision

precision = precision_score(y_test, y_pred, average='weighted')

# Calculate recall

recall = recall_score(y_test, y_pred, average='weighted')

print("Accuracy : ",accuracy)
print("Precision : ",precision)
print('Recall : ',recall)

import numpy as np

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_curve, auc

from sklearn.preprocessing import label_binarize

```

```

# Convert y_test to binary labels
y_test_binary = label_binarize(y_test, classes=np.unique(y_test))

# Calculate the scores for each class
y_pred_scores = nb_classifier.predict_proba(X_test_scaled)

# Compute the false positive rate (fpr), true positive rate (tpr), and area under the ROC curve
(auc) for each class

fpr = dict()
tpr = dict()
roc_auc = dict()

n_classes = y_test_binary.shape[1]

for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_binary[:, i], y_pred_scores[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curve for each class

plt.figure()

for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f)' % roc_auc[i])

    plt.plot([0, 1], [0, 1], 'k--') # Diagonal line

    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])

```

```
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Receiver Operating Characteristic (ROC) Curve')  
plt.legend(loc="lower right")  
plt.show()
```

add Codeadd Markdown

Random Forest

```
from sklearn.ensemble import RandomForestClassifier  
  
from sklearn.metrics import classification_report
```

Train the Random Forest model

```
random_forest_model = RandomForestClassifier()  
  
random_forest_model.fit(X_train, y_train)
```

Evaluate the model

```
y_pred = random_forest_model.predict(X_test)  
  
report = classification_report(y_test, y_pred)  
  
print(report)
```

Calculate accuracy

```
accuracy = accuracy_score(y_test, y_pred)
```

Calculate precision

```
precision = precision_score(y_test, y_pred, average='weighted')
```

```

# Calculate recall

recall = recall_score(y_test, y_pred, average='weighted')

print("Accuracy : ",accuracy)
print("Precision : ",precision)
print('Recall : ',recall)

import numpy as np

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_curve, auc

from sklearn.preprocessing import label_binarize

# Convert y_test to binary labels

y_test_binary = label_binarize(y_test, classes=np.unique(y_test))

# Calculate the scores for each class

y_pred_scores = random_forest_model.predict_proba(X_test)

# Compute the false positive rate (fpr), true positive rate (tpr), and area under the ROC curve
(auc) for each class

fpr = dict()
tpr = dict()
roc_auc = dict()

n_classes = y_test_binary.shape[1]

```

```

for i in range(n_classes):

    fpr[i], tpr[i], _ = roc_curve(y_test_binary[:, i], y_pred_scores[:, i])

    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curve for each class

plt.figure()

for i in range(n_classes):

    plt.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f)' % roc_auc[i])

plt.plot([0, 1], [0, 1], 'k--') # Diagonal line

plt.xlim([0.0, 1.0])

plt.ylim()

# data analysis and manipulation libraries

import numpy as np

import pandas as pd

from datetime import timedelta

# data visualization libraries

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from wordcloud import WordCloud

# libraries for nlp tasks

import nltk

import re

from nltk.tokenize import word_tokenize

```

```

from nltk.stem import WordNetLemmatizer

# Model used for Sentiment Analysis

from nltk.sentiment import SentimentIntensityAnalyzer

# For topic modelling

import gensim

from sklearn.feature_extraction.text import CountVectorizer

import pyLDAvis.gensim

from gensim.corpora.dictionary import Dictionary

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.decomposition import PCA

# for storing objects as pickle files

import pickle

# Counting items in a list and returning as a dictionary

from collections import Counter

# library for statistical analysis

from scipy import stats

# Disable all warnings

import warnings

warnings.filterwarnings("ignore")

#import imaging Library

from PIL import Image

add Codeadd Markdown

# importing the dataset

df = pd.read_csv("Twitter Jan Mar.csv")

def convert_to_datetime(x):

```

```
try:
```

```
    return pd.to_datetime(x)
```

```
except:
```

```
    return np.nan
```

```
df['date'] = df['date'].apply(lambda x: convert_to_datetime(x))
```

```
print("Null values Count, Prior to any Processing:")
```

```
print(df.isnull().sum())
```

```
df = df.dropna()
```

```
df["date only"] = df["date"].dt.date
```

```
df = df.drop("id", axis=1) #id not providing any useful information
```

```
print("Null values Count, After Processing:")
```

```
print(df.isnull().sum())
```

```
df.head(5)
```

Text processing for Sentiment analysis

For the Sentiment Analysis of the tweets, we have decided to use the VADER(Valence Aware Dictionary for Sentiment Reasoning) which has been designed for Social Media Sentiment Analysis. Another advantage it possesses is that it does not require a lot of preprocessing. However, the text data will be processed to use for Topic Modelling later.

- Pre-processing for Sentiment Analysis: hashtag, url, account mention removal
- Pre-processing for LDA: same as Sentiment analysis + expand contractions, remove punctuations and emoticons, remove stop words, lowercase all text

```
# Extract out hashtags in the tweets
```

```
df["accounts_mentioned"] = df["content"].apply(lambda x: re.findall("(?=<^|(?=<[^a-zA-Z0-9-_\\.])@([A-Za-z]+[A-Za-z0-9_]+)", x) )
```

```
# Extract out mentioned accounts in the tweets
```

```

df["hashtags"] = df["content"].apply(lambda x: re.findall("#(\w+)",x))

tweets = df["content"] #df with only tweets

# remove urls

tweets = tweets.apply(lambda x: re.sub('http\S+', " ", x))

# remove mentions

tweets = tweets.apply(lambda x: re.sub("(?<=^|(?<=[^a-zA-Z0-9-_\.])@([A-Za-z]+[A-Za-z0-9_]+)", " ", x))

# remove hashtags

tweets = tweets.apply(lambda x: re.sub("#(\w+)", " ", x))

# expand contractions

import contractions

tweets = tweets.apply(lambda x: contractions.fix(x))

# remove punctuations and emoticons

tweets = tweets.apply(lambda x: re.sub('[^\w\s]', " ", x))

# remove stop words

from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

tweets = tweets.apply(lambda x: " ".join([w for w in x.split() if w.lower() not in stop_words]) )

# lower case all words

tweets = tweets.apply(lambda x: x.lower() )

df = pd.merge(df, tweets, how="inner", left_index=True, right_index=True)

# lemmatize the words after tokenizing

df['content_y'] = df['content_y'].apply(lambda x: word_tokenize(x))

lemmatizer = WordNetLemmatizer()

```

```

df['content_y'] = df['content_y'].apply(lambda x: " ".join([lemmatizer.lemmatize(i) for i in
x]) )

# remove hashtags, urls and accounts mentioned

df["content_to_input"] = df["content_x"]

df["content_to_input"] = df["content_to_input"].apply(lambda x: re.sub('http\S+', " ", x))

df["content_to_input"] = df["content_to_input"].apply(lambda x: re.sub("(?<=^|(?<=[^a-zA-Z0-9-_\\.))@[A-Za-z]+[A-Za-z0-9_]+)", " ", x))

df["content_to_input"] = df["content_to_input"].apply(lambda x: re.sub("#(\w+)", " ", x))

df.head(5)

ax = sns.histplot(data=df, x="Sentiment_Label", stat="percent" )

plt.title("Sentiment Distribution")

plt.ylabel("Distribution (percentage)")

ax.spines[['right', 'top']].set_visible(False)

```

add Codeadd Markdown

Divide the dataframe into 3 child dataframes based on sentiment.

```

df_pos = df[df["Sentiment_Label"]=="Positive"]

df_neu = df[df["Sentiment_Label"]=="Neutral"]

df_neg = df[df["Sentiment_Label"]=="Negative"]

```

add Codeadd Markdown

Trend in Tweet Sentiment

Anomalies

```

df_grouped_date_sentiment = df.groupby(by=["date
only","Sentiment_Label"],as_index=False).count()

df_grouped_date = df.groupby(by=["date only"],as_index=False).count()

```

```

df_grouped_date = df_grouped_date[["date only","content_x"]]

df_grouped_date_sentiment = df_grouped_date_sentiment[["date
only","Sentiment_Label","content_x"]]

max_2 = df_grouped_date.sort_values(by="content_x",ascending=False).iloc[:2]

fig = plt.subplots(2,1,figsize=(13,12))

plt.subplot(2,1,1)

sns.lineplot(data=df_grouped_date , x="date only",y="content_x")

sns.scatterplot(data=df_grouped_date , x="date only",y="content_x")

plt.ylabel("No. of Tweets")

plt.xlabel("Date")

plt.title("Overall Trend in Tweet Count")

plt.text(max_2["date only"].iloc[1]+timedelta(days=1), max_2["content_x"].iloc[1],
str(max_2["date only"].iloc[1]))

plt.text(max_2["date only"].iloc[0]+timedelta(days=1), max_2["content_x"].iloc[0],
str(max_2["date only"].iloc[0]))

plt.plot(max_2["date only"].iloc[0], max_2["content_x"].iloc[0],marker='o')

plt.plot(max_2["date only"].iloc[1], max_2["content_x"].iloc[1],marker='o')

plt.tight_layout()

plt.subplot(2,1,2)

sns.lineplot(data=df_grouped_date_sentiment , x="date only",y="content_x",
hue="Sentiment_Label", legend="full")

plt.ylabel("No. of Tweets")

plt.xlabel("Date")

plt.title("Trend in Tweet Count overlayed by Sentiment")

plt.tight_layout()

```

add Codeadd Markdown

Seasonality

add Codeadd Markdown

Hourly Seasonality

There is an observable trend in the number of tweets that grow from 0 to 13 hours and plateaus till 18 hours and then decreases with time.

```
time =df[["date", 'date only', "Sentiment_Label"]]
```

```
time["hour of day"] = time["date"].dt.hour
```

```
# pandas df representing number of tweets every hour grouped by day and sentiment label.
```

```
hourly = time.groupby(by=["date only",'hour of
```

```
day',"Sentiment_Label"],as_index=False).count()
```

```
hourly[\"Tweet Count\"] =hourly[\"date\"]
```

```
hourly_sentiment = hourly.drop(["date","date only"], axis=1)
```

```
# group by df without sentiment label
```

```
hourly_non_sentiment = time.groupby(by=["date only",'hour of
```

```
day'],as_index=False).count()
```

```
hourly_non_sentiment [\"Tweet Count\"] =hourly_non_sentiment [\"date\"]
```

```
hourly_non_sentiment = hourly_non_sentiment.drop(["date","date only","Sentiment_Label"],  
axis=1)
```

```
fig, axes = plt.subplots(1,2,figsize=(18,7))
```

```
plt.suptitle("Average Tweet Distribution Based on Hours in a Day", size=22)
```

```
plt.subplot(1,2,1)
```

```
ax1 = sns.barplot(data=hourly_non_sentiment, x='hour of day', y="Tweet Count",  
color="violet")
```

```
plt.axvspan(13, 18, color='purple', alpha=0.1)
```

```

plt.ylabel("Avg. Tweets")

plt.subplot(1,2,2)

ax2 = sns.lineplot(data=hourly_sentiment, x='hour of day', y="Tweet Count",
hue="Sentiment_Label")

plt.ylabel("Avg. Tweets")

plt.show()

```

add Codeadd Markdown

Days of the Week Seasonality

Twitter activity tends to be relatively lower during weekends, compared to week days.

```

time =df[["date", 'date only', "Sentiment_Label"]]

time["week day"] = time["date"].dt.day_of_week

# pandas df representing number of tweets every hour grouped by day and sentiment label.

week_day = time.groupby(by=["date only",'week
day',"Sentiment_Label"],as_index=False).count()

week_day["Tweet Count"] =week_day["date"]

week_day = week_day.drop(["date","date only"], axis=1)

#week_day_sentiment = week_day.groupby(by=['week
day',"Sentiment_Label"],as_index=False).mean()

# group by df without sentiment label

week_day_non_sentiment = time.groupby(by=["date only",'week
day'],as_index=False).count()

week_day_non_sentiment ["Tweet Count"] =week_day_non_sentiment ["date"]

week_day_non_sentiment = week_day_non_sentiment.drop(["date","date
only","Sentiment_Label"], axis=1)

```

```

#week_day_non_sentiment = week_day_non_sentiment.groupby(by=['week
day'],as_index=False).mean()

fig, axes = plt.subplots(1,2,figsize=(18,7))

plt.suptitle("Average Tweet Distribution Based on Days of the Week", size=22)

plt.subplot(1,2,1)

ax1 = sns.barplot(data=week_day_non_sentiment , x='week day', y="Tweet Count",
color="violet")

plt.xticks([0, 1, 2, 3, 4, 5, 6],
["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"])

plt.ylabel("Avg. Tweets")

plt.subplot(1,2,2)

ax2 = sns.lineplot(data=week_day , x='week day', y="Tweet Count",
hue="Sentiment_Label")

plt.xticks([0, 1, 2, 3, 4, 5, 6],
["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"])

plt.ylabel("Avg. Tweets")

plt.show()

```

```

extra_words = ["chat", "gpt", "ai", "artificial", "intelligence", "chatgpt", "gpt4"]

b = x.split()

c = []

for i in b:

    if i.lower() in extra_words:

        continue

    c.append(i)

```

```

return " ".join(c)

df_7feb = df[df["date only"] == pd.to_datetime("2023-02-07").date() ]

df_7feb["content_y"] = df_7feb["content_y"].apply(remove_extra_words)

long_text = " ".join([i for i in df_7feb["content_y"].values])

w_cloud = WordCloud(background_color="white", max_words=3000, contour_width=5,
contour_color='steelblue')

w_cloud.generate(long_text)

plt.imshow(w_cloud, interpolation='bilinear')

plt.axis('off')

df_15mar = df[df["date only"] == pd.to_datetime("2023-03-15").date() ]

df_15mar["content_y"] = df_15mar["content_y"].apply(remove_extra_words)

long_text = " ".join([i for i in df_15mar["content_y"].values])

w_cloud = WordCloud(background_color="white", max_words=3000, contour_width=5,
contour_color='steelblue')

w_cloud.generate(long_text)

plt.imshow(w_cloud, interpolation='bilinear')

plt.axis('off')

```

add Codeadd Markdown

Topic Modelling based on Sentiment

```

pos_tweet = df_pos["content_y"]

neg_tweet = df_neg["content_y"]

neu_tweet = df_neu["content_y"]

# remove all words containg chat, gpt, ai, artificial, intelligence

def remove_extra_words(x):

    extra_words = ["chat", "gpt", "ai", "artificial", "intelligence", "chatgpt", "gpt4"]

```

```

b = x.split()
c = []
for i in b:
    if i.lower() in extra_words:
        continue
    c.append(i)
return " ".join(c)

pos_tweet = pos_tweet.apply(lambda x: remove_extra_words(x))
neg_tweet = neg_tweet.apply(lambda x: remove_extra_words(x))
neu_tweet = neu_tweet.apply(lambda x: remove_extra_words(x))

# remove words less than 3 letters in length

def rless3(x):
    b = x.split()
    c = []
    for i in b:
        if len(i)<=3:
            continue
        c.append(i)
    return " ".join(c)

pos_tweet = pos_tweet.apply(lambda x: rless3(x))
neg_tweet = neg_tweet.apply(lambda x: rless3(x))
neu_tweet = neu_tweet.apply(lambda x: rless3(x))

# Count vectorization

vect_pos = CountVectorizer(min_df=20, ngram_range = (1,1))
vect_neg = CountVectorizer(min_df=20, ngram_range = (1,1))

```

```

vect_neu = CountVectorizer(min_df=20, ngram_range = (1,1))

# Fit and transform

X_pos = vect_pos.fit_transform(pos_tweet)

X_neg = vect_neg.fit_transform(neg_tweet)

X_neu = vect_neu.fit_transform(neu_tweet)

# Convert sparse matrix to gensim corpus.

corpus_pos = gensim.matutils.Sparse2Corpus(X_pos, documents_columns=False)

corpus_neg = gensim.matutils.Sparse2Corpus(X_neg, documents_columns=False)

corpus_neu = gensim.matutils.Sparse2Corpus(X_neu, documents_columns=False)

# Mapping from word IDs to words (To be used in LdaModel's id2word parameter)

id_map_pos = dict((v, k) for k, v in vect_pos.vocabulary_.items())

id_map_neg = dict((v, k) for k, v in vect_neg.vocabulary_.items())

id_map_neu = dict((v, k) for k, v in vect_neu.vocabulary_.items())

# Training LDA Models

topic_num_pos = 4

topic_num_neg = 3

topic_num_neu = 3

ldamodel_pos = gensim.models.ldamodel.LdaModel(corpus=corpus_pos,
                                                id2word=id_map_pos,num_topics=topic_num_pos, random_state=23,passes=15)

ldamodel_neg = gensim.models.ldamodel.LdaModel(corpus=corpus_neg,
                                                id2word=id_map_neg,num_topics=topic_num_neg, random_state=3,passes=15)

ldamodel_neu = gensim.models.ldamodel.LdaModel(corpus=corpus_neu,
                                                id2word=id_map_neu,num_topics=topic_num_neu, random_state=5,passes=15)

dict = { "pos": [ldamodel_pos, vect_pos] , "neg": [ldamodel_neg, vect_neg] ,
         "neu": [ldamodel_neu, vect_neu] }

def topic_model(x, sentiment):

```

```

x_l = []
x_l.append(x)
x_vect = dict[sentiment][1].transform(x_l)

corpus = gensim.matutils.Sparse2Corpus(x_vect, documents_columns=False)

topic = dict[sentiment][0].get_document_topics(corpus)

max_p = 0

mp_topic = 0

for i in topic[0]:
    if i[1]>max_p:
        max_p = i[1]
        mp_topic = i[0]

return mp_topic

```

```

pos_tweet_df = pd.DataFrame(pos_tweet)

neg_tweet_df = pd.DataFrame(neg_tweet)

neu_tweet_df = pd.DataFrame(neu_tweet)

pos_tweet_df["Topic"] = pos_tweet_df["content_y"].apply(lambda x: topic_model(x,"pos"))

neg_tweet_df["Topic"] = neg_tweet_df["content_y"].apply(lambda x: topic_model(x,"neg"))

neu_tweet_df["Topic"] = neu_tweet_df["content_y"].apply(lambda x: topic_model(x,"neu"))

add Codeadd Markdown

```

Possible Topics in Postive Tweets

```

corpus_dict = Dictionary.from_corpus(corpus_pos, id2word=ldamodel_pos.id2word)

vis = pyLDAvis.gensim.prepare(ldamodel_pos, corpus_pos, dictionary=corpus_dict )

pyLDAvis.save_html(vis, 'lda_pos.html')# Can be viewed in html file

```

```

topics_pos = {0:"Interaction of \n humans with ChatGPT", 1:"Usefulness in \n Education",
2:"LLMs and Search \n Engines", 3:"Potenitals of AI \n in the future" }

pos_tweet_df["Topic"] = pos_tweet_df["Topic"].apply(lambda x: topics_pos[x])

plt.figure(figsize=(14,8))

ax1 = sns.histplot(data=pos_tweet_df, x="Topic", stat="percent" )

plt.tight_layout()

plt.ylabel("")

plt.xlabel("Positive Tweet Topics")

rects = ax1.patches

for rect in rects:

    height = rect.get_height()

    ax1.text(
        rect.get_x() + rect.get_width() / 2, height, str(height)[:5] + " %", ha="center",
        va="bottom"
    )

corpus_dict = Dictionary.from_corpus(corpus_neg, id2word=ldamodel_neg.id2word)

vis = pyLDAvis.gensim.prepare(ldamodel_neg, corpus_neg, dictionary=corpus_dict )

pyLDAvis.save_html(vis, 'lda_neg.html')# Can be viewed in html file

Image.open("contact_neg.jpg")

topics_neg = {0:"It's Limitations", 1:"Regarding using \n ChatGPT for \n writing tasks",
2:"Risks involved \n during the \n use of ChatGPT" }

neg_tweet_df["Topic"] = neg_tweet_df["Topic"].apply(lambda x: topics_neg[x])

plt.figure(figsize=(14,8))

ax2 = sns.histplot(data=neg_tweet_df, x="Topic", stat="percent" )

plt.tight_layout()

plt.ylabel("")

```

```

plt.xlabel("Negative Tweet Topics")

rects = ax2.patches

for rect in rects:

    height = rect.get_height()

    ax2.text(
        rect.get_x() + rect.get_width() / 2, height, str(height)[:5] + " %", ha="center",
        va="bottom"
    )

add Codeadd Markdown

corpus_dict = Dictionary.from_corpus(corpus_neu, id2word=ldamodel_neu.id2word)

vis = pyLDAvis.gensim.prepare(ldamodel_neu, corpus_neu, dictionary=corpus_dict )

pyLDAvis.save_html(vis, 'lda_neu.html')# Can be viewed in html file

Image.open("contact_neu.jpg")

add Codeadd Markdown

Distribution of the Topics in Tweets:

topics_neu = {0:"Generation \n of Ideas", 1:"Advancements in \n AI and its Impacts",
2:"Upcoming plans for \n LLMs and investments"}

neu_tweet_df["Topic"] = neu_tweet_df["Topic"].apply(lambda x: topics_neu[x])

plt.figure(figsize=(14,8))

ax3 = sns.histplot(data=neu_tweet_df, x="Topic", stat="percent" )

plt.tight_layout()

plt.ylabel("")

plt.xlabel("Neutral Tweet Topics")

rects = ax3.patches

for rect in rects:

```

```
height = rect.get_height()

ax3.text(
    rect.get_x() + rect.get_width() / 2, height, str(height)[:5] + " %", ha="center",
    va="bottom"
)
```

add Codeadd Markdown

Appendix B: Paper LSTM with RNN implementation

```
!wget https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.vec.gz
```

```
!gunzip "/content/cc.en.300.vec.gz"
```

```
from gensim.models import KeyedVectors
```

```
filepath = "cc.en.300.vec"
w2v = KeyedVectors.load_word2vec_format(filepath, binary=False)
print(len(w2v.vocab))
```

```
MAX_SEQUENCE_LENGTH = 85 #176 # based on our inputs; TODO: remove outliers?
dynamically calculate!
```

```
MAX_NB_WORDS = len(w2v.vocab)
EMBEDDING_DIM = 300 # w2v, fastText; GloVe=50
```

```
sample = w2v["Hassan"]
print(sample.shape)
print(w2v.most_similar("Hassan"))
```

```
import pandas as pd
import numpy as np
```

```
import tensorflow as tf

df = pd.read_csv('train.csv')
df.head()

conditions = [
    (df['sentiment'] == 'positive'),
    (df['sentiment'] == 'negative'),
    (df['sentiment'] == 'neutral'),
]
values = [0, 1, 2]
classes = ["positive", "negative", "neutral"]
df['sentiment_classes'] = np.select(conditions, values)
df.head()

df.drop('selected_text', 1, inplace=True)
df.drop('sentiment', 1, inplace=True)
df.drop('textID', 1, inplace=True)
df.head()

df.dropna(axis=0, inplace=True)

texts = list(df.text)
labels = list(df.sentiment_classes)

print('Found %s texts.' % len(texts))
print(texts[0], labels[0])

from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
# from keras.utils import to_categorical
from tensorflow.keras.utils import to_categorical
```

```

MAX_NUM_WORDS = 200

# finally, vectorize the text samples into a 2D integer tensor
tokenizer = Tokenizer(num_words=MAX_NUM_WORDS)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

print("sequence***", len(sequences[0]), sequences[0])
word_index = tokenizer.word_index

print('Found %s unique tokens.' % len(word_index))

MAX_SEQUENCE_LENGTH = 1000

data = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

labels = to_categorical(np.asarray(labels))
print('Shape of data tensor:', data.shape)
print('Shape of label tensor:', labels.shape)

from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(data, labels, test_size=0.2, random_state=42)
x_test, x_val, y_test, y_val = train_test_split(data, labels, test_size=0.5, random_state=42)

EMBEDDING_DIM = 300

print('Preparing embedding matrix.')

# prepare embedding matrix
num_words = min(MAX_NUM_WORDS, len(word_index) + 1)
embedding_matrix = np.zeros((num_words, EMBEDDING_DIM))

```

```

for word, i in word_index.items():
    if i >= MAX_NUM_WORDS:
        continue
    if word in w2v.vocab:
        embedding_vector = w2v[word]
        embedding_vector = np.array(embedding_vector)
        if embedding_vector is not None:
            # words not found in embedding index will be all-zeros.
            embedding_matrix[i] = embedding_vector
print(embedding_matrix.shape)

```

```

from keras.models import Sequential
from keras.layers import Dense, Embedding, Bidirectional, LSTM
from keras.layers import Flatten
from keras.initializers import Constant

print('Training model.')

```

```

# define the model
model = Sequential()
model.add(Embedding(num_words,
                    EMBEDDING_DIM,
                    embeddings_initializer=Constant(embedding_matrix),
                    input_length=MAX_SEQUENCE_LENGTH,
                    trainable=False))

```

```

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(tf.keras.layers.Dropout(0.1))
model.add(Dense(256, activation='relu'))
model.add(tf.keras.layers.Dropout(0.1))
model.add(Dense(128, activation='relu'))

```

```
model.add(tf.keras.layers.Dropout(0.1))

model.add(Dense(3, activation='softmax'))

# compile the model
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['acc'])

# summarize the model
model.summary()

history = model.fit(x_train, y_train,
                     batch_size=512,
                     epochs=30,
                     validation_data=(x_val, y_val))

loss, accuracy = model.evaluate(x_test, y_test, verbose=1)

print('Accuracy: %f % (accuracy)')
print('Loss: %f % (loss)')

import matplotlib.pyplot as plt
acc = history.history['acc']
val_acc = history.history['val_acc']

loss=history.history['loss']
val_loss=history.history['val_loss']

#epochs_range = range(22)

plt.figure(figsize=(15, 15))
plt.subplot(1, 2, 1)
```

```

plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()

```

Appendix Paper C: Paper “Sentiment Analysis Predictions in Digital Media Content using NLP Techniques”

```

import numpy as np
import pandas as pd
data = pd.read_csv('/content/drive/My Drive/Data Visualization/Internet Calling.csv')
data.head()
reviews = np.array(data['Text'])[:1000]
labels = np.array(data['Sentiment'])[:1000]

len(reviews)

data['Text'].loc[139]

data['Sentiment'].loc[139]

```

```
from collections import Counter
```

```
Counter(labels)
```

```
punctuation = '!"#$%&\'()*+,-.:;<=>?[\\]^_`{|}~'
```

```
# get rid of punctuation
```

```
all_reviews = 'separator'.join(reviews)
```

```
all_reviews = all_reviews.lower()
```

```
all_text = ".join([c for c in all_reviews if c not in punctuation])
```

```
# split by new lines and spaces
```

```
reviews_split = all_text.split('separator')
```

```
all_text = ''.join(reviews_split)
```

```
# create a list of words
```

```
words = all_text.split()
```

```
# get rid of web address, twitter id, and digit
```

```
new_reviews = []
```

```
for review in reviews_split:
```

```
    review = review.split()
```

```
    new_text = []
```

```
for word in review:
```

```
    if (word[0] != '@') & ('http' not in word) & (~word.isdigit()):
```

```
        new_text.append(word)
```

```
new_reviews.append(new_text)
```

```
## Build a dictionary that maps words to integers
```

```
counts = Counter(words)
```

```
vocab = sorted(counts, key=counts.get, reverse=True)
```

```
vocab_to_int = {word: ii for ii, word in enumerate(vocab, 1)}
```

```
## use the dict to tokenize each review in reviews_split
```

```
## store the tokenized reviews in reviews_ints
```

```
reviews_ints = []
```

```
for review in new_reviews:
```

```
    reviews_ints.append([vocab_to_int[word] for word in review])
```

```
# stats about vocabulary
```

```
print('Unique words: ', len((vocab_to_int))) # should ~ 74000+
```

```
print()
```

```
# print tokens in first review
```

```
print('Tokenized review: \n', reviews_ints[:1])
```

```
# 1=positive, 1=neutral, 0=negative label conversion
```

```
encoded_labels = []
```

```

for label in labels:

    if label == 'neutral':
        encoded_labels.append(1)

    elif label == 'negative':
        encoded_labels.append(0)

    else:
        encoded_labels.append(1)

encoded_labels = np.asarray(encoded_labels)

def pad_features(reviews_ints, seq_length):

    """ Return features of review_ints, where each review is padded with 0's
    or truncated to the input seq_length.

    """

# getting the correct rows x cols shape

features = np.zeros((len(reviews_ints), seq_length), dtype=int)

# for each review, I grab that review and

for i, row in enumerate(reviews_ints):

    features[i, -len(row):] = np.array(row)[:seq_length]

return features

# Test implementation!

```

```
seq_length = 30

features = pad_features(reviews_ints, seq_length=seq_length)

## test statements

assert len(features)==len(reviews_ints), "The features should have as many rows as reviews."
assert len(features[0])==seq_length, "Each feature row should contain seq_length values."

# print first 10 values of the first 30 batches
print(features[:10,:10])

split_frac = 0.8

## split data into training, validation, and test data (features and labels, x and y)

split_idx = int(len(features)*split_frac)
train_x, remaining_x = features[:split_idx], features[split_idx:]
train_y, remaining_y = encoded_labels[:split_idx], encoded_labels[split_idx:]

test_idx = int(len(remaining_x)*0.5)
val_x, test_x = remaining_x[:test_idx], remaining_x[test_idx:]
val_y, test_y = remaining_y[:test_idx], remaining_y[test_idx:]

## print out the shapes of the resultant feature data
```

```

print("\t\tFeature Shapes:")
print("Train set: \t\t{ }".format(train_x.shape),
"\nValidation set: \t\t{ }".format(val_x.shape),
"\nTest set: \t\t{ }".format(test_x.shape))

import torch

from torch.utils.data import TensorDataset, DataLoader

# create Tensor datasets

train_data = TensorDataset(torch.from_numpy(train_x), torch.from_numpy(train_y))
valid_data = TensorDataset(torch.from_numpy(val_x), torch.from_numpy(val_y))
test_data = TensorDataset(torch.from_numpy(test_x), torch.from_numpy(test_y))

# dataloaders

batch_size = 50

# make sure the SHUFFLE the training data

train_loader = DataLoader(train_data, shuffle=True, batch_size=batch_size)
valid_loader = DataLoader(valid_data, shuffle=True, batch_size=batch_size)
test_loader = DataLoader(test_data, shuffle=True, batch_size=batch_size)

# obtain one batch of training data

dataiter = iter(train_loader)
sample_x, sample_y = dataiter.next()

```

```
print('Sample input size: ', sample_x.size()) # batch_size, seq_length  
print('Sample input: \n', sample_x)  
print()  
print('Sample label size: ', sample_y.size()) # batch_size  
print('Sample label: \n', sample_y)
```

First checking if GPU is available

```
train_on_gpu=torch.cuda.is_available()
```

```
if(train_on_gpu):
```

```
    print('Training on GPU.')
```

```
else:
```

```
    print('No GPU available, training on CPU.')
```

```
import torch.nn as nn
```

```
class SentimentRNN(nn.Module):
```

```
    """
```

The RNN model that will be used to perform Sentiment analysis.

```
    """
```

```
def __init__(self, vocab_size, output_size, embedding_dim, hidden_dim, n_layers,  
drop_prob=0.5):
```

```
    """
```

Initialize the model by setting up the layers.

```

"""
super(SentimentRNN, self).__init__()

self.output_size = output_size
self.n_layers = n_layers
self.hidden_dim = hidden_dim

# embedding and LSTM layers
self.embedding = nn.Embedding(vocab_size, embedding_dim)
self.lstm = nn.LSTM(embedding_dim, hidden_dim, n_layers,
                   dropout=drop_prob, batch_first=True)

# dropout layer
self.dropout = nn.Dropout(0.3)

# linear and sigmoid layers
self.fc = nn.Linear(hidden_dim, output_size)
self.sig = nn.Sigmoid()

def forward(self, x, hidden):
"""

Perform a forward pass of our model on some input and hidden state.

batch_size = x.size(0)

```

```

# embeddings and lstm_out

x = x.long()

embeds = self.embedding(x)

lstm_out, hidden = self.lstm(embeds, hidden)

# stack up lstm outputs

lstm_out = lstm_out.contiguous().view(-1, self.hidden_dim)

# dropout and fully-connected layer

out = self.dropout(lstm_out)

out = self.fc(out)

# sigmoid function

sig_out = self.sig(out)

# reshape to be batch_size first

sig_out = sig_out.view(batch_size, -1)

sig_out = sig_out[:, -1] # get last batch of labels

# return last sigmoid output and hidden state

return sig_out, hidden

def init_hidden(self, batch_size):

    """ Initializes hidden state """

```

```

# Create two new tensors with sizes n_layers x batch_size x hidden_dim,
# initialized to zero, for hidden state and cell state of LSTM

weight = next(self.parameters()).data

if (train_on_gpu):
    hidden = (weight.new(self.n_layers, batch_size, self.hidden_dim).zero_().cuda(),
              weight.new(self.n_layers, batch_size, self.hidden_dim).zero_().cuda())
else:
    hidden = (weight.new(self.n_layers, batch_size, self.hidden_dim).zero_(),
              weight.new(self.n_layers, batch_size, self.hidden_dim).zero_())

return hidden

# Instantiate the model w/ hyperparams

vocab_size = len(vocab_to_int)+1 # +1 for the 0 padding + our word tokens
output_size = 1
embedding_dim = 200
hidden_dim = 128
n_layers = 2

net = SentimentRNN(vocab_size, output_size, embedding_dim, hidden_dim, n_layers)

print(net)

# loss and optimization functions

```

```
lr=0.001
```

```
criterion = nn.BCELoss()
```

```
optimizer = torch.optim.Adam(net.parameters(), lr=lr)
```

```
# training params
```

```
epochs = 10
```

```
counter = 0
```

```
print_every = 100
```

```
clip=5 # gradient clipping
```

```
# move model to GPU, if available
```

```
if(train_on_gpu):
```

```
    net.cuda()
```

```
net.train()
```

```
# train for some number of epochs
```

```
for e in range(epochs):
```

```
    # initialize hidden state
```

```
    h = net.init_hidden(batch_size)
```

```
    # batch loop
```

```
    for inputs, labels in train_loader:
```

```

counter += 1

if(train_on_gpu):
    inputs, labels = inputs.cuda(), labels.cuda()

# Creating new variables for the hidden state, otherwise
# we'd backprop through the entire training history
h = tuple([each.data for each in h])

# zero accumulated gradients
net.zero_grad()

# get the output from the model
output, h = net(inputs, h)

# calculate the loss and perform backprop
loss = criterion(output.squeeze(), labels.float())
loss.backward()

# `clip_grad_norm` helps prevent the exploding gradient problem in RNNs / LSTMs.
nn.utils.clip_grad_norm_(net.parameters(), clip)
optimizer.step()

# loss stats
if counter % print_every == 0:
    # Get validation loss

```

```

val_h = net.init_hidden(batch_size)

val_losses = []

net.eval()

for inputs, labels in valid_loader:

    # Creating new variables for the hidden state, otherwise
    # we'd backprop through the entire training history

    val_h = tuple([each.data for each in val_h])

    if(train_on_gpu):

        inputs, labels = inputs.cuda(), labels.cuda()

        output, val_h = net(inputs, val_h)

        val_loss = criterion(output.squeeze(), labels.float())

        val_losses.append(val_loss.item())

    net.train()

    print("Epoch: { }/{ }...".format(e+1, epochs),

          "Step: { }...".format(counter),

          "Loss: {:.6f}...".format(loss.item()),

          "Val Loss: {:.6f}{}".format(np.mean(val_losses)))

# Get test data loss and accuracy

```

```

test_losses = [] # track loss

num_correct = 0


# init hidden state

h = net.init_hidden(batch_size)

net.eval()

# iterate over test data

for inputs, labels in test_loader:

    # Creating new variables for the hidden state, otherwise
    # we'd backprop through the entire training history

    h = tuple([each.data for each in h])

    if(train_on_gpu):

        inputs, labels = inputs.cuda(), labels.cuda()

    # get predicted outputs

    output, h = net(inputs, h)

    # calculate loss

    test_loss = criterion(output.squeeze(), labels.float())

    test_losses.append(test_loss.item())


    # convert output probabilities to predicted class (0 or 1)

```

```

pred = torch.round(output.squeeze()) # rounds to the nearest integer

# compare predictions to true label
correct_tensor = pred.eq(labels.float().view_as(pred))

correct = np.squeeze(correct_tensor.numpy()) if not train_on_gpu else
np.squeeze(correct_tensor.cpu().numpy())

num_correct += np.sum(correct)

# -- stats! -- ##
# avg test loss
print("Test loss: {:.3f}".format(np.mean(test_losses)))

# accuracy over all test data
test_acc = num_correct/len(test_loader.dataset)
print("Test accuracy: {:.3f}".format(test_acc))

# negative test review
test_review = " Any other suggestions than zoom so I can make my commitment??""

def tokenize_review(test_review):
    test_review = test_review.lower() # lowercase
    # get rid of punctuation
    test_text = ''.join([c for c in test_review if c not in punctuation])

```

```

# splitting by spaces

test_words = test_text.split()

# get rid of web address, twitter id, and digit

new_text = []

for word in test_words:

    if (word[0] != '@') & ('http' not in word) & (~word.isdigit()):

        new_text.append(word)

# tokens

test_ints = []

test_ints.append([vocab_to_int[word] for word in new_text])



return test_ints


# test code and generate tokenized review

test_ints = tokenize_review(test_review)

print(test_ints)


# test sequence padding

seq_length=30

features = pad_features(test_ints, seq_length)


print(features)

```

```
# test conversion to tensor and pass into your model

feature_tensor = torch.from_numpy(features)

print(feature_tensor.size())

def predict(net, test_review, sequence_length=30):

    net.eval()

    # tokenize review

    test_ints = tokenize_review(test_review)

    # pad tokenized sequence

    seq_length=sequence_length

    features = pad_features(test_ints, seq_length)

    # convert to tensor to pass into your model

    feature_tensor = torch.from_numpy(features)

    batch_size = feature_tensor.size(0)

    # initialize hidden state

    h = net.init_hidden(batch_size)

    if(train_on_gpu):
```

```

feature_tensor = feature_tensor.cuda()

# get the output from the model

output, h = net(feature_tensor, h)

# convert output probabilities to predicted class (0 or 1)

pred = torch.round(output.squeeze())

# printing output value, before rounding

print('Prediction value, pre-rounding: {:.6f}'.format(output.item()))

# print custom response

if(pred.item()==1):

    print("Non-negative review detected.")

else:

    print("Negative review detected.")

seq_length = 30 # good to use the length that was trained on

# call function on negative review

test_review_neg = "you have my money, Any other suggestions so I can make my commitment??"

predict(net, test_review_neg, seq_length)

# call function on positive review

test_review_pos = "@India thank you we got on a different vision to world."

```

```
predict(net, test_review_pos, seq_length)

# call function on neutral review

test_review_neu = "i need someone to help me out"

predict(net, test_review_neu, seq_length)

import pickle

model_pkl_file='sentimental_model.pkl'

with open(model_pkl_file,'wb')as file:

    pickle.dump(SentimentRNN(vocab_size, output_size, embedding_dim, hidden_dim,
n_layers),file)

with open(model_pkl_file, 'rb') as file:

    pickle_model = pickle.load(file)

pickle_model.train

df = pd.read_csv('./train.csv', encoding = 'latin')

df = df.dropna()

df.head()

remove_text = "@\|S+|https?:\|S+|http?:\|S|[^\w\d]+"
```

```
stop_words = stopwords.words("english")

stemmer = SnowballStemmer("english")

def preprocess(text, stem=False):

    text = re.sub(remove_text, ' ', str(text).lower()).strip()

    tokens = []

    for token in text.split():

        if token not in stop_words:

            if stem:

                tokens.append(stemmer.stem(token))

            else:

                tokens.append(token)

    return " ".join(tokens)

df.text = df.text.apply(lambda x: preprocess(x))

df.sentiment = df.sentiment.map({ "neutral": 1, "negative":0, "positive":2 })

df.head()

msg = df['text']
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.cluster import KMeans

from sklearn.metrics import adjusted_rand_score

vectorizer = TfidfVectorizer()

X = vectorizer.fit_transform(msg)

tf_idf = pd.DataFrame(data = X.toarray(), columns=vectorizer.get_feature_names())

final_df = tf_idf

print("{} rows".format(final_df.shape[0]))

#final_df.T.nlargest(5, 0)

def run_KMeans(max_k, data):

    max_k += 1

    kmeans_results = dict()

    for k in range(2 , max_k):

        kmeans = KMeans(n_clusters = k

                        , init = 'k-means++'
```

```

        , n_init = 10

        , tol = 0.0001

        , n_jobs = -1

        , random_state = 1

        , algorithm = 'full')

kmeans_results.update( {k : kmeans.fit(data)} )

return kmeans_results

def printAvg(avg_dict):

    for avg in sorted(avg_dict.keys(), reverse=True):

        print("Avg: {} \tK:{} ".format(avg.round(4), avg_dict[avg]))

def plotSilhouette(df, n_clusters, kmeans_labels, silhouette_avg):

    fig, ax1 = plt.subplots(1)

    fig.set_size_inches(8, 6)

    ax1.set_xlim([-0.2, 1])

    ax1.set_ylim([0, len(df) + (n_clusters + 1) * 10])

```

```

ax1.axvline(x=silhouette_avg, color="red", linestyle="--") # The vertical line for average
silhouette score of all the values

ax1.set_yticks([]) # Clear the yaxis labels / ticks

ax1.set_xticks([-0.2, 0, 0.2, 0.4, 0.6, 0.8, 1])

plt.title(("Silhouette analysis for K = %d" % n_clusters), fontsize=10, fontweight='bold')

y_lower = 10

sample_silhouette_values = silhouette_samples(df, kmeans_labels) # Compute the
silhouette scores for each sample

for i in range(n_clusters):

    ith_cluster_silhouette_values = sample_silhouette_values[kmeans_labels == i]

    ith_cluster_silhouette_values.sort()

    size_cluster_i = ith_cluster_silhouette_values.shape[0]

    y_upper = y_lower + size_cluster_i

    color = cm.nipy_spectral(float(i) / n_clusters)

    ax1.fill_betweenx(np.arange(y_lower, y_upper), 0, ith_cluster_silhouette_values,
                     facecolor=color, edgecolor=color, alpha=0.7)

```

```

ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i)) # Label the silhouette plots with
their cluster numbers at the middle

y_lower = y_upper + 10 # Compute the new y_lower for next plot. 10 for the 0 samples

plt.show()

def silhouette(kmeans_dict, df, plot=False):

    df = df.to_numpy()

    avg_dict = dict()

    for n_clusters, kmeans in kmeans_dict.items():

        kmeans_labels = kmeans.predict(df)

        silhouette_avg = silhouette_score(df, kmeans_labels) # Average Score for all Samples

        avg_dict.update( {silhouette_avg : n_clusters} )

    if(plot): plotSilhouette(df, n_clusters, kmeans_labels, silhouette_avg)

# Running Kmeans

k = 3

kmeans_results = run_KMeans(k, final_df)

```

```

# Plotting Silhouette Analysis

#silhouette(kmeans_results, final_df, plot=True)

def get_top_features_cluster(tf_idf_array, prediction, n_feats):

    labels = np.unique(prediction)

    dfs = []

    for label in labels:

        id_temp = np.where(prediction==label) # indices for each cluster

        x_means = np.mean(tf_idf_array[id_temp], axis = 0) # returns average score across
        cluster

        sorted_means = np.argsort(x_means)[::-1][:n_feats] # indices with top 20 scores

        features = vectorizer.get_feature_names()

        best_features = [(features[i], x_means[i]) for i in sorted_means]

        df = pd.DataFrame(best_features, columns = ['features', 'score'])

        dfs.append(df)

    return dfs

def plotWords(dfs, n_feats):

    plt.figure(figsize=(8, 4))

    for i in range(0, len(dfs)):

        df = dfs[i]
        cluster_label = str(i)
        plt.title('Top ' + str(n_feats) + ' Features for Cluster ' + cluster_label)

        plt.xlabel('Score')
        plt.ylabel('Feature')

        plt.bar(df['features'], df['score'])

        plt.show()

```

```
plt.title(("Most Common Words in Cluster { }".format(i)), fontsize=10,
fontweight='bold')

sns.barplot(x = 'score' , y = 'features', orient = 'h' , data = dfs[i][:n_feats])

plt.show()

import matplotlib.pyplot as plt

import seaborn as sns

best_result = 3

kmeans = kmeans_results.get(best_result)

final_df_array = final_df.to_numpy()

prediction = kmeans.predict(final_df)

n_feats = 20

dfs = get_top_features_cluster(final_df_array, prediction, n_feats)

plotWords(dfs, 13)

# Transforms a centroids dataframe into a dictionary to be used on a WordCloud.

def centroidsDict(centroids, index):

    a = centroids.T[index].sort_values(ascending = False).reset_index().values
```

```
centroid_dict = dict()

for i in range(0, len(a)):

    centroid_dict.update( {a[i,0] : a[i,1]} )

return centroid_dict

def generateWordClouds(centroids):

    wordcloud = WordCloud(max_font_size=100, background_color = 'white')

    for i in range(0, len(centroids)):

        centroid_dict = centroidsDict(centroids, i)

        wordcloud.generate_from_frequencies(centroid_dict)

        plt.figure()

        plt.title('Cluster {}'.format(i))

        plt.imshow(wordcloud)

        plt.axis("off")

        plt.show()

from wordcloud import WordCloud
```

```
centroids = pd.DataFrame(kmeans.cluster_centers_)

centroids.columns = final_df.columns

generateWordClouds(centroids)

from sklearn.decomposition import PCA

kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 0)

kmean_indices = kmeans.fit_predict(final_df)

pca = PCA(n_components=2)

scatter_plot_points = pca.fit_transform(final_df)

colors = ["red", "blue", "green"]

x_axis = [o[0] for o in scatter_plot_points]

y_axis = [o[1] for o in scatter_plot_points]

fig, ax = plt.subplots(figsize=(20,10))

ax.scatter(x_axis, y_axis, c=[colors[d] for d in kmean_indices])

#ax.legend()

df['cluster'] = kmeans.labels_
```

```
df.head()
```

```
df3 = df[['text', 'cluster']]
```

```
df3.head()
```

```
twt = df['text']
```

```
sent = df['sentiment']
```

```
x_train, x_test, y_train, y_test = train_test_split(twt, sent, test_size = 0.2, random_state = 0)
```

```
train = []
```

```
test = []
```

```
for i in x_train.index:
```

```
    temp=x_train[i]
```

```
    train.append(temp)
```

```
for j in x_test.index:
```

```
    temp1=x_test[j]
```

```
test.append(temp1)

cv = CountVectorizer()

x_train = cv.fit_transform(x_train)

x_test = cv.transform(x_test)

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive_bayes import MultinomialNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import LinearSVC

from sklearn.ensemble import VotingClassifier

from sklearn.metrics import accuracy_score

knn = KNeighborsClassifier(n_neighbors=3)

knn.fit(x_train,y_train)

#After training we test the algorithm on test data.

y_pred_knn = knn.predict(x_test)

acc = accuracy_score(y_test, y_pred_knn)
```

acc

nb = MultinomialNB()

nb.fit(x_train,y_train)

y_pred_nb = nb.predict(x_test)

acc = accuracy_score(y_test, y_pred_nb)

acc

dt = DecisionTreeClassifier(random_state=0)

dt.fit(x_train,y_train)

y_pred_dt=dt.predict(x_test)

acc = accuracy_score(y_test, y_pred_dt)

acc

forest = RandomForestClassifier(n_estimators=500, min_samples_leaf=2)

forest.fit(x_train,y_train)

y_pred_rf = forest.predict(x_test)

acc = accuracy_score(y_test, y_pred_rf)

acc

```
svc=LinearSVC(random_state= 0 ,max_iter=15000)

svc.fit(x_train,y_train)

y_pred_svc=svc.predict(x_test)

acc = accuracy_score(y_test, y_pred_svc)

acc

estimator = []

estimator.append(('KNN', KNeighborsClassifier(n_neighbors=3)))

estimator.append(('MNB', MultinomialNB()))

estimator.append(('DTC', DecisionTreeClassifier(random_state=0)))

estimator.append(('SVC', LinearSVC(random_state= 0 ,max_iter=15000)))

estimator.append(('RFC', RandomForestClassifier(n_estimators=500, min_samples_leaf=2)))

vot_hard = VotingClassifier(estimators = estimator, voting ='hard')

vot_hard.fit(x_train, y_train)

y_pred_hard = vot_hard.predict(x_test)

acc = accuracy_score(y_test, y_pred_hard)

acc

## LSTM
```

```
from keras.models import Sequential  
  
from keras.preprocessing.text import Tokenizer  
  
from keras.preprocessing.sequence import pad_sequences  
  
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
```

```
df = pd.read_csv('./train.csv')
```

```
df.head()
```

```
def preprocess(message):
```

```
    """
```

This function takes a string as input, then performs these operations:

- lowercase
- remove URLs
- remove ticker symbols
- removes punctuation
- tokenize by splitting the string on whitespace
- removes any single character tokens

Parameters

message : The text message to be preprocessed.

Returns

tokens: The preprocessed text into tokens.

Lowercase the twit message

```
text = str(message).lower()
```

Replace URLs with a space in the message

```
text = re.sub(r'https?:\/\/.*[\r\n]*', ' ', text)
```

Replace ticker symbols with a space. The ticker symbols are any stock symbol that starts with \$.

```
text = re.sub(r'[$][A-Za-z][\S]*', ' ', text)
```

Replace StockTwits usernames with a space. The usernames are any word that starts with @.

```
text = re.sub(r'@[A-Za-z][\S]*', ' ', text)
```

```
# Replace everything not a letter with a space
```

```
text = re.sub(r'[\W_]+', ' ', text)
```

```
# Tokenize by splitting the string on whitespace into a list of words
```

```
tokens = text.split()
```

```
# Lemmatize words using the WordNetLemmatizer. You can ignore any word that is not  
longer than one character.
```

```
wnl = nltk.stem.WordNetLemmatizer()
```

```
tokens = [wnl.lemmatize(token) for token in tokens if len(token) > 1]
```

```
return tokens
```

```
df.text = df.text.apply(lambda x: preprocess(x))
```

```
df.sentiment = df.sentiment.map({"neutral": 1, "negative":0, "positive":2 })
```

```
df.head()
```

```
tokenizer = Tokenizer(num_words=500, split=' ')
```

```
tokenizer.fit_on_texts(df['text'].values)

X = tokenizer.texts_to_sequences(df['text'].values)

X = pad_sequences(X)

y=pd.get_dummies(df['sentiment'])

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2, random_state = 42)

model = Sequential()

model.add(Embedding(500, 120, input_length = X.shape[1]))

model.add(SpatialDropout1D(0.4))

model.add(LSTM(500, dropout=0.2, recurrent_dropout=0.2))

model.add(Dense(3,activation='softmax'))

model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])

print(model.summary())

batch_size=64

model.fit(X_train, y_train, epochs = 30, batch_size=batch_size, validation_data=(X_test,
y_test), verbose = 'auto')

import pandas as pd
```

```
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

import torch

from torch.utils.data import Dataset

from transformers import DistilBertTokenizerFast,DistilBertForSequenceClassification

from transformers import Trainer,TrainingArguments

train_texts = train['text'].values.tolist()

train_labels = train['sentiment'].values.tolist()

train_texts, val_texts, train_labels, val_labels = train_test_split(train_texts, train_labels, test_size=.2,random_state=42,stratify=train_labels)

model_name = 'distilbert-base-uncased'

tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-base-uncased',num_labels=3)

train_encodings = tokenizer(train_texts, truncation=True, padding=True,return_tensors = 'pt')
```

```
val_encodings = tokenizer(val_texts, truncation=True, padding=True, return_tensors = 'pt')

class SentimentDataset(torch.utils.data.Dataset):

    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
        item['labels'] = torch.tensor(self.labels[idx])
        return item

    def __len__(self):
        return len(self.labels)

def compute_metrics(p):
    pred, labels = p
    pred = np.argmax(pred, axis=1)
    accuracy = accuracy_score(y_true=labels, y_pred=pred)
```

```
return {"accuracy": accuracy}

training_args = TrainingArguments(
    output_dir='./res',      # output directory
    evaluation_strategy="steps",
    num_train_epochs=15,      # total number of training epochs
    per_device_train_batch_size=64, # batch size per device during training
    per_device_eval_batch_size=64, # batch size for evaluation
    warmup_steps=500,         # number of warmup steps for learning rate scheduler
    weight_decay=0.01,        # strength of weight decay
    logging_dir='./logs4',    # directory for storing logs
    #logging_steps=10,
    load_best_model_at_end=True,
)

model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-
uncased",num_labels=3)

trainer = Trainer(
    model=model,# the instantiated Transformers model to be trained
```

```

args=training_args, # training arguments, defined above

train_dataset=train_dataset,# training dataset

eval_dataset=val_dataset , # evaluation dataset

compute_metrics=compute_metrics,


)

trainer.train()

```

Appendix Paper D: Conference Paper “A Novel Approach to Predict the Real Time Sentimental Analysis by Naive Bayes & RNN Algorithm during the COVID Pandemic in UAE”

```

from tweepy import OAuthHandler

from tweepy.streaming import StreamListener

import tweepy

import json

import pandas as pd

import csv

import re

from textblob import TextBlob

import string

#import preprocessor as p

import os

import time

from datetime import datetime

```

```

from retrying import retry

from tweepy import Stream

import json

import seaborn as sns

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from nltk.stem import PorterStemmer

from nltk.stem import WordNetLemmatizer

# ML Libraries

from sklearn.metrics import accuracy_score

from sklearn.naive_bayes import MultinomialNB

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC


# Global Parameters

stop_words = set(stopwords.words('english'))


# Twitter credentials

consumer_key = 'UWUE5DUi8I0FCzXTmJIetHvjT'

consumer_key_secret = 'Ajn5CKRYc4X8SVNEUbTjcP0Hkvi4arEPJeWjBlQtjizzMw6Oo9'

access_token = '712590332966273024-yeLdqwE9CSmlQ7V8TUB8UYN0Ihj3yUu'

access_token_secret = 'xTWhbEn9jN2cI0gN7yI0u5Sk3luqbX3WeSRg6ixMRzUb0'

auth = tweepy.OAuthHandler(consumer_key, consumer_key_secret)

```

```

auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

search_words = ["india"]

date_since = "2019-01-21"

number_of_tweets=1000

tweets = tweepy.Cursor(api.search,q=search_words,
                       geocode="20.5937,78.9629,300km",
                       lang="en", since=date_since).items(number_of_tweets)

#tweets=api.search(search_words,geocode="20.5937,78.9629,300km",lang="en",
since=date_since,count=1000)

## the geocode is for India; format for geocode="latitude,longitude,radius"
## radius should be in miles or km

data_tweets=[]

for tweet in tweets:

    dict={"Created_At":tweet.created_at,"User_Name":tweet.user.screen_name,"Text":tweet.text
          ,"Location":tweet.user.location}

    data_tweets.append(dict)

print (data_tweets)

tweet_df=pd.DataFrame(data_tweets,columns=['Created_At','User_Name','Text','Location'])

print('Dataset size:',tweet_df.shape)

```

```
print('Columns are:',tweet_df.columns)
```

```
tweet_df.info()
```

```
tweet_df.dropna(subset=['Text'],inplace=True)
```

```
tweet_df.isnull().sum()
```

```
def remove_punct(text):  
  
    text = "".join([char for char in text if char not in string.punctuation])  
  
    text = re.sub('[0-9]+', " ", text)  
  
    return text
```

```
tweet_df['Tweet_punct'] = tweet_df['Text'].apply(lambda x: remove_punct(x))
```

```
tweet_df.head(10)
```

```
import pandas as pd
```

```
import numpy as np
```

```
import json
```

```
import re
```

```
import string
```

```
import nltk
```

```
nltk.download('punkt')
```

```
nltk.download('stopwords')
```

```
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize


def tokenization(text):

    text = re.split('\W+', text)

    return text


tweet_df['Tweet_tokenized'] = tweet_df['Tweet_punct'].apply(lambda x:
    tokenization(x.lower()))

tweet_df.head()

stopword = nltk.corpus.stopwords.words('english')

def remove_stopwords(text):

    text = [word for word in text if word not in stopword]

    return text


tweet_df['Tweet_nonstop'] = tweet_df['Tweet_tokenized'].apply(lambda x:
    remove_stopwords(x))

tweet_df.head(10)

ps = nltk.PorterStemmer()

def stemming(text):
```

```

text = [ps.stem(word) for word in text]

return text


tweet_df['Tweet_stemmed'] = tweet_df['Tweet_nonstop'].apply(lambda x: stemming(x))

tweet_df.head()


nltk.download('wordnet')

wn = nltk.WordNetLemmatizer()


def lemmatizer(text):

    text = [wn.lemmatize(word) for word in text]

    return text


tweet_df['Tweet_lemmatized'] = tweet_df['Tweet_nonstop'].apply(lambda x: lemmatizer(x))

tweet_df.head()


def clean_text(text):

    text_lc = "".join([word.lower() for word in text if word not in string.punctuation]) #

    remove punctuation

    text_rc = re.sub('[0-9]+', ' ', text_lc)

    tokens = re.split('\W+', text_rc) # tokenization

    text = [ps.stem(word) for word in tokens if word not in stopword] # remove stopwords and

    stemming

    return text

```

```

from sklearn.feature_extraction.text import CountVectorizer
countVectorizer = CountVectorizer(analyzer=clean_text)
countVector = countVectorizer.fit_transform(tweet_df['Text'])

print('{} Number of tweets has {} words'.format(countVector.shape[0],
countVector.shape[1]))

count_vect_df = pd.DataFrame(countVector.toarray(),
columns=countVectorizer.get_feature_names())
count_vect_df.head()

def remove_pattern(text,pattern):
    # re.findall() finds the pattern i.e @user and puts it in a list for further task
    r = re.findall(pattern,text)

    # re.sub() removes @user from the sentences in the dataset
    for i in r:
        text = re.sub(i,"",text)

    return text

tweet_df["Tidy_Tweets"] = np.vectorize(remove_pattern)(tweet_df["Text"], "@[\w]*")

tweet_df.head()

```

```
tweet_df['Tidy_Tweets'] = tweet_df['Tidy_Tweets'].apply(lambda x: ''.join([w for w in x.split() if len(w)>3]))
```

```
tweet_df.head(10)
```

```
from textblob import TextBlob
```

```
def clean_tweet(tweet):
```

```
    """
```

```
        Utility function to clean tweet text by removing links, special characters
```

```
        using simple regex statements.
```

```
    """
```

```
    return ''.join(re.sub("@[A-Za-z0-9]+|[^0-9A-Za-z \t]|(\w+:\/\/[\S+]", " ",
```

```
tweet).split())
```

```
def get_tweet_sentiment(tweet):
```

```
    """
```

```
        Utility function to classify sentiment of passed tweet
```

```
        using textblob's sentiment method
```

```
    """
```

```
# create TextBlob object of passed tweet text
```

```
analysis = TextBlob(clean_tweet(tweet))
```

```
# set sentiment
```

```
if analysis.sentiment.polarity > 0:
```

```
    return 'positive'
```

```
elif analysis.sentiment.polarity == 0:
```

```
    return 'neutral'
```

```

else:
    return 'negative'

tweet_df['Sentiment'] = tweet_df['Text'].apply(lambda x: get_tweet_sentiment(x))

from wordcloud import WordCloud,ImageColorGenerator
from PIL import Image
import urllib
import requests

all_words_positive = ''.join(text for text in
    tweet_df['Tidy_Tweets'][tweet_df['Sentiment']=='positive'])

# combining the image with the dataset
Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-
PNG-Image.png', stream=True).raw))

# We use the ImageColorGenerator library from Wordcloud
# Here we take the color of the image and impose it over our wordcloud
image_colors = ImageColorGenerator(Mask)
wc = WordCloud(background_color='black', height=1500,
width=4000,mask=Mask).generate(all_words_positive)

import matplotlib.pyplot as plt

```

```

# Size of the image generated
plt.figure(figsize=(10,20))

# Here we recolor the words from the dataset to the image's color
# recolor just recolors the default colors to the image's blue color
# interpolation is used to smooth the image generated
plt.imshow(wc.recolor(color_func=image_colors),interpolation="hamming")

plt.axis('off')
plt.show()

all_words_negative = ' '.join(text for text in
tweet_df["Tidy_Tweets"][tweet_df['Sentiment']=='negative'])

# combining the image with the dataset
Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-
PNG-Image.png', stream=True).raw))

# We use the ImageColorGenerator library from Wordcloud
# Here we take the color of the image and impose it over our wordcloud
image_colors = ImageColorGenerator(Mask)

# Now we use the WordCloud function from the wordcloud library

```

```

wc = WordCloud(background_color='black', height=1500,
width=4000,mask=Mask).generate(all_words_negative)

# Size of the image generated
plt.figure(figsize=(10,20))

# Here we recolor the words from the dataset to the image's color
# recolor just recolors the default colors to the image's blue color
# interpolation is used to smooth the image generated
plt.imshow(wc.recolor(color_func=image_colors),interpolation="gaussian")

plt.axis('off')
plt.show()

```

```
def Hashtags_Extract(x):
```

```

    hashtags=[]

    # Loop over the words in the tweet
    for i in x:
        ht = re.findall(r'#[\w+]',i)
        hashtags.append(ht)

```

```
return hashtags
```

```
ht_positive = Hashtags_Extract(tweet_df['Text'][tweet_df['Sentiment']=='positive'])
```

```
ht_positive
```

```
ht_positive_unnest = sum(ht_positive,[])
```

```
ht_negative = Hashtags_Extract(tweet_df['Text'][tweet_df['Sentiment']=='negative'])
```

```
ht_negative
```

```
ht_negative_unnest = sum(ht_negative,[])
```

```
word_freq_positive = nltk.FreqDist(ht_positive_unnest)
```

```
word_freq_positive
```

```
df_positive =
```

```
pd.DataFrame({'Hashtags':list(word_freq_positive.keys()),'Count':list(word_freq_positive.values())})
```

```
df_positive.head(10)
```

```
import seaborn as sns
```

```
df_positive_plot = df_positive.nlargest(20,columns='Count')
```

```
sns.barplot(data=df_positive_plot,y='Hashtags',x='Count')
```

```
sns.despine()
```

```
word_freq_negative = nltk.FreqDist(ht_negative_unnest)
```

```
word_freq_negative
```

```
df_negative =
```

```
pd.DataFrame({'Hashtags':list(word_freq_negative.keys()),'Count':list(word_freq_negative.v  
alues())})
```

```
df_negative.head(10)
```

```
df_negative_plot = df_negative.nlargest(20,columns='Count')
```

```
sns.barplot(data=df_negative_plot,y='Hashtags',x='Count')
```

```
sns.despine()
```

```
def Get_label(tweet):
```

```
    if tweet=='positive':
```

```
        return 4
```

```
    elif tweet=='negative':
```

```
        return 0
```

```
    else:
```

```
        return 2
```

```
tweet_df['label_1'] = tweet_df['Sentiment'].apply(lambda x: Get_label(x))
```

```
tweet_df.columns
```

```
tweet_df.label_1.value_counts()
```

```
sns.countplot(x= 'label_1',data = tweet_df)
```

```
def load_dataset(filename, cols):
```

```
    dataset = pd.read_csv(filename, encoding='latin-1')
```

```
    dataset.columns = cols
```

```
    return dataset
```

```
def remove_unwanted_cols(dataset, cols):
```

```
    for col in cols:
```

```
        del dataset[col]
```

```
    return dataset
```

```
def preprocess_tweet_text(tweet):
```

```
    tweet.lower()
```

```
    # Remove urls
```

```
    tweet = re.sub(r"http\S+|www\S+|https\S+", " ", tweet, flags=re.MULTILINE)
```

```

# Remove user @ references and '#' from tweet

tweet = re.sub(r'@\w+|\#', '', tweet)

# Remove punctuations

tweet = tweet.translate(str.maketrans("", "", string.punctuation))

# Remove stopwords

tweet_tokens = word_tokenize(tweet)

filtered_words = [w for w in tweet_tokens if not w in stop_words]

#ps = PorterStemmer()

#stemmed_words = [ps.stem(w) for w in filtered_words]

#lemmatizer = WordNetLemmatizer()

#lemma_words = [lemmatizer.lemmatize(w, pos='a') for w in stemmed_words]

return " ".join(filtered_words)

def get_feature_vector(train_fit):

    vector = TfidfVectorizer(sublinear_tf=True)

    vector.fit(train_fit)

    return vector

def int_to_string(sentiment):

    if sentiment == 0:

        return "Negative"

    elif sentiment == 2:

        return "Neutral"

```

```

else:
    return "Positive"

tweet_df.to_csv('training.csv',index=False)

tweet_df.columns

dataset = load_dataset("training.csv", ['Created_At', 'User_Name', 'Text', 'Location',
'Tweet_punct',
'Tweet_tokenized', 'Tweet_nonstop', 'Tweet_stemmed', 'Tweet_lemmatized',
'Tidy_Tweets', 'Sentiment','label_1'])

n_dataset = remove_unwanted_cols(dataset,
['Created_At','User_Name','Location','Tweet_punct',
'Tweet_tokenized', 'Tweet_nonstop', 'Tweet_stemmed', 'Tweet_lemmatized',
'Tidy_Tweets', 'Sentiment',])

print (dataset.columns)

dataset.text = dataset['Text'].apply(preprocess_tweet_text)

dataset.columns

# Split dataset into Train, Test

# Same tf vector will be used for Testing sentiments on unseen trending data

```

```
tf_vector = get_feature_vector(np.array(dataset.iloc[:,0]).ravel())

X = tf_vector.transform(np.array(dataset.iloc[:,0]).ravel())

y = np.array(dataset.iloc[:, 1]).ravel()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=30)
```

```
# Training Naive Bayes model

NB_model = MultinomialNB()

NB_model.fit(X_train, y_train)

y_predict_nb = NB_model.predict(X_test)

print(accuracy_score(y_test, y_predict_nb))
```

```
# Training Logistics Regression model

LR_model = LogisticRegression(solver='lbfgs')

LR_model.fit(X_train, y_train)

y_predict_lr = LR_model.predict(X_test)

print(accuracy_score(y_test, y_predict_lr))
```

```
import speech_recognition as sr

import moviepy.editor as mp

def convert_vid_to_audio(videofile):

    clip = mp.VideoFileClip(videofile).subclip(0,20)

    clip.audio.write_audiofile("spkr0.wav")
```

```
def extract_text(audiofile):
    r = sr.Recognizer()

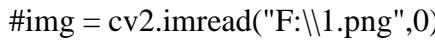
    with sr.AudioFile(audiofile) as source:
        #reads the audio file. Here we use record instead of
        #listen
        audio = r.record(source)

        #print("The audio file contains: " + r.recognize_google(audio))

    try:
        audio_gathered=r.recognize_google(audio)
        return audio_gathered
    except:
        return None

from matplotlib import pyplot as plt
from PIL import Image
import pytesseract
import argparse
import cv2
import os
```

```


#img = cv2.imread("F:\\1.png",0)

def imagetotext(image):
    img = cv2.imread(image,0)
    plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
    plt.xticks([]), plt.yticks([]) # to hide tick values on X and Y axis
    plt.show()
    filename = "{}.png".format(os.getpid())
    cv2.imwrite(filename, img)
    pytesseract.pytesseract.tesseract_cmd = 'F:\\Google\\tesseract'
    text = pytesseract.image_to_string(Image.open(filename))
    return text

```

```

from io import StringIO

def sentimental_analysis(text):
    StringData = StringIO("""text;
    {}
""".format(text))

    test_ds = pd.read_csv(StringData, sep =";",index_col=False)

    # Creating text feature
    test_ds.text = test_ds["text"].apply(preprocess_tweet_text)

    test_feature = tf_vector.transform(np.array(test_ds.iloc[:, 0]).ravel())

# Using Logistic Regression model for prediction
test_prediction_lr = NB_model.predict(test_feature)

```

```

# Averaging out the hashtags result

test_result = pd.DataFrame({'prediction':test_prediction_lr})

test_result.columns = ['predictions']

test_result.predictions = test_result['predictions'].apply(int_to_string)

print(test_result)

print ('\t 1.Text Analysis \
\t 2.Audio Analysis \
\t 3.Image Analysis')

choice=int(input('Enter your Choice'))

if choice==1:

    string_input=input('Enter your text')

    sentimental_analysis(string_input)

elif choice==2:

    audio_file_path=input('Enter the path for Audio File')

    if os.path.isfile(audio_file_path):

        print ("Conversion Begin")

        output=extract_text(audio_file_path)

        print (output)

        sentimental_analysis (output)

    else:

        print ('Please check the path')

else:

```

```

image_file_path=input('Enter the path of the Image File')

if os.path.isfile(image_file_path):

    print ('Extracting Text from Image')

    output=imagetotext(image_file_path)

    sentimental_analysis (output)

else:

    print ('Image file not exists')


from geopy.geocoders import Nominatim

geolocator = Nominatim()

city ="London"

country ="Uk"

loc = geolocator.geocode(city+','+ country)

print("latitude is :-" ,loc.latitude,"\\longitude is:-" ,loc.longitude)

tweet_df.columns

def change_delimiter(location):

    tweet=re.sub('W+',',',location)

    return tweet


tweet_df['ChangedLocation']=tweet_df.Location.apply(lambda x:

change_delimiter(x))

tweet_location=tweet_df.ChangedLocation.replace(r",India,India")

```

```
def latandlong(location):
    loc = geolocator.geocode(location)
    if loc:
        return (loc.latitude, loc.longitude)
    else:
        return None

tweet_df['latandlong']=tweet_df.ChangedLocation.head(20).apply(lambda
x:latandlong(x))

tweet_df.columns

def converttimestamptodate(date):
    date=re.findall('\d{4}\-\d{2}\-\d{2}',str(date))
    return date[0]

tweet_df['onlydate']=tweet_df.Created_At.apply(lambda
x:converttimestamptodate(x))

tweet_df.onlydate.head()

date=list(set(tweet_df.onlydate))

count=list(set(tweet_df.onlydate.value_counts()))
```

```
plt.plot(tweet_df.onlydate)
```

```
tweet_df.columns
```

```
labels=['Negative','Neutral','Positive']  
size=list(set(tweet_df.label_1.value_counts()))
```

```
colors = ['gold', 'yellowgreen', 'lightcoral']  
#explode = (0.1, 0, 0, 0) # explode 1st slice
```

```
# Plot
```

```
patches, texts = plt.pie(size, colors=colors, shadow=True, startangle=90)  
plt.legend(patches, labels, loc="best")  
plt.axis('equal')  
plt.tight_layout()  
plt.show()
```

```
def latlongseperation(tweet):
```

```
    tweet=tweet.split(',')  
    return(tweet[0],tweet[1])
```

```
lantandlong=list(set(tweet_df.latandlong.head(20)))
```

```
lat=[]
```

```
long=[]

for i in lantandlong:

    if i is not None:

        lat.append(i[0])

        long.append(i[1])
```

lat

long

```
import gmplot

gmap4 = gmplot.GoogleMapPlotter.from_geocode("India")
```

Appendix Paper E: Paper “Use of Artificial Intelligence Applications in order to learn the Sentiment Polarity: A Case Study of the Public Perceptions on the Organizations Providing Post COVID-19 Vaccinations in the UAE”

Importing Libraries

```
import os

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from nltk.corpus import stopwords
```

```
from nltk.stem import WordNetLemmatizer  
from nltk.tokenize import word_tokenize  
from wordcloud import WordCloud,STOPWORDS  
from bs4 import BeautifulSoup  
import re,string,unicodedata  
  
  
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer  
from sklearn.linear_model import LogisticRegression  
from sklearn.svm import LinearSVC  
from sklearn.ensemble import GradientBoostingClassifier  
from sklearn.naive_bayes import GaussianNB, MultinomialNB  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix,  
plot_confusion_matrix, plot_roc_curve, plot_precision_recall_curve  
from xgboost.sklearn import XGBClassifier  
  
  
import tensorflow as tf  
from tensorflow.keras.preprocessing.text import Tokenizer  
from tensorflow.keras.preprocessing.sequence import pad_sequences  
from tensorflow.keras.callbacks import ModelCheckpoint  
from tensorflow.keras.layers import Dense, Input, Embedding, LSTM, Dropout, Conv1D,  
MaxPooling1D, GlobalMaxPooling1D, Dropout, Bidirectional, Flatten, BatchNormalization  
from tensorflow.keras.callbacks import EarlyStopping  
from tensorflow.keras.models import Model  
from tensorflow.keras.optimizers import Adam
```

```

from tensorflow.keras.utils import plot_model
import transformers
import tokenizers

Loading Dataset

data=pd.read_csv('../input/imdb-dataset-of-50k-movie-reviews/IMDB Dataset.csv')

data.head()

data['review'][10000:15000]

data.describe() #descriptive statistics

num_duplicates = data.duplicated().sum() #identify duplicates

print('There are {} duplicate reviews present in the dataset'.format(num_duplicates))

#drop duplicate reviews

data.drop_duplicates(inplace = True)

print('The dataset contains {} rows and {} columns after removing
duplicates'.format(data.shape[0],data.shape[1]))

```

Data Pre-processing

```

stop = stopwords.words('english')

wl = WordNetLemmatizer()

#mapping

mapping = {"ain't": "is not", "aren't": "are not", "can't": "cannot",
           "'cause": "because", "could've": "could have", "couldn't": "could not",
           "didn't": "did not", "doesn't": "does not", "don't": "do not", "hadn't": "had not",
           "hasn't": "has not", "haven't": "have not", "he'd": "he would", "he'll": "he will",
           }

```

"he's": "he is", "how'd": "how did", "how'd'y": "how do you", "how'll": "how will",
"how's": "how is", "I'd": "I would", "I'd've": "I would have", "I'll": "I will",
"I'll've": "I will have", "I'm": "I am", "I've": "I have", "i'd": "i would",
"i'd've": "i would have", "i'll": "i will", "i'll've": "i will have",
"i'm": "i am", "i've": "i have", "isn't": "is not", "it'd": "it
would",
"it'd've": "it would have", "it'll": "it will", "it'll've": "it will have",
"it's": "it is", "let's": "let us", "ma'am": "madam", "mayn't": "may not",
"might've": "might have", "mightn't": "might not", "mightn't've": "might not have",
"must've": "must have", "mustn't": "must not", "mustn't've": "must not have",
"needn't": "need not", "needn't've": "need not have", "o'clock": "of the clock",
"oughtn't": "ought not", "oughtn't've": "ought not have", "shan't": "shall not",
"sha'n't": "shall not", "shan't've": "shall not have", "she'd": "she would",
"she'd've": "she would have", "she'll": "she will", "she'll've": "she will have",
"she's": "she is", "should've": "should have", "shouldn't": "should not",
"shouldn't've": "should not have", "so've": "so have", "so's": "so as", "this's": "this is",
"that'd": "that would", "that'd've": "that would have", "that's": "that is",
"there'd": "there would", "there'd've": "there would have", "there's": "there is",
"here's": "here is", "they'd": "they would", "they'd've": "they would have",
"they'll": "they will", "they'll've": "they will have", "they're": "they are",
"they've": "they have", "to've": "to have", "wasn't": "was not", "we'd": "we would",
"we'd've": "we would have", "we'll": "we will", "we'll've": "we will have",
"we're": "we are", "we've": "we have", "weren't": "were not",
"what'll": "what will", "what'll've": "what will have", "what're": "what are",

"what's": "what is", "what've": "what have", "when's": "when is", "when've": "when have",
 "where'd": "where did", "where's": "where is", "where've": "where have", "who'll":
 "who will",
 "who'll've": "who will have", "who's": "who is", "who've": "who have", "why's": "why
 is",
 "why've": "why have", "will've": "will have", "won't": "will not", "won't've": "will not
 have",
 "would've": "would have", "wouldn't": "would not", "wouldn't've": "would not have",
 "y'all": "you all", "y'all'd": "you all would", "y'all'd've": "you all would have",
 "y'all're": "you all are", "y'all've": "you all have", "you'd": "you would",
 "you'd've": "you would have", "you'll": "you will", "you'll've": "you will have",
 "you're": "you are", "you've": "you have" }

#data cleaning

```

def preprocess_text(text, lemmatize = True):
    soup = BeautifulSoup(text, "html.parser") #remove html tags
    text = soup.get_text()
    text = ''.join([mapping[t] if t in mapping else t for t in text.split(" ")]) #expanding
    chatwords and contracts clearing contractions
    emoji_clean= re.compile("["
        u"\U0001F600-\U0001F64F" # emoticons
        u"\U0001F300-\U0001F5FF" # symbols & pictographs
        u"\U0001F680-\U0001F6FF" # transport & map symbols
    ]
  
```

```

u"\U0001F1E0-\U0001F1FF" #flags (iOS)

u"\U00002702-\U000027B0"

u"\U000024C2-\U0001F251"

"]+", flags=re.UNICODE)

text = emoji_clean.sub(r'',text)

text = re.sub(r'\.(?=|S)', ' ',text) #add space after full stop

text = re.sub(r'httplib\S+', " ", text) #remove urls

text = """.join([word.lower() for word in text if word not in string.punctuation]) #remove punctuation

#tokens = re.split('\W+', text) #create tokens

if lemmatize:

    text = " ".join([wl.lemmatize(word) for word in text.split() if word not in stop and
word.isalpha()]) #lemmatize

else:

    text = " ".join([word for word in text.split() if word not in stop and word.isalpha()])

return text

data_copy = data.copy()

data['review']=data['review'].apply(preprocess_text, lemmatize = True)

#converting target variable to numeric labels

data.sentiment = [ 1 if each == "positive" else 0 for each in data.sentiment]

#after converting labels

data.head()

```

Opinion Mining and Sentiment Classification

#Plot sentiments

```
import seaborn as sns  
  
sns.set(style='darkgrid')  
  
sns.set(font_scale=1.2)  
  
sns.countplot(data=data, x='sentiment', palette=['orange', 'blue'], order=[1, 0])  
  
plt.xticks(ticks=[0, 1], labels=['positive', 'negative'])  
  
plt.title('Opinion count from the dataset')  
  
plt.show()
```

#print percentages

```
positive_count = data['sentiment'].value_counts()[0]  
  
positive_percent = round(positive_count/len(data)*100,2)
```

```
negative_count = data['sentiment'].value_counts()[1]
```

```
negative_percent = round(negative_count/len(data)*100,2)
```

#output percentages

```
print('Positive opinions are',positive_count, 'i.e.',positive_percent,'% of IMDB dataset')  
  
print('Negative opinions are',negative_count, 'i.e.',negative_percent,'% of IMDB dataset')
```

#create pie chart

```
labels = ['Positive', 'Negative']  
  
sizes = [positive_percent, negative_percent]  
  
colors = ['orange', 'blue']
```

```
plt.pie(sizes,labels=labels,colors=colors, autopct='%.1f%%')

plt.axis('equal')

plt.title("Distribution of sentiments")

plt.show()
```

#Word cloud for positive opinions

```
word_cloud = WordCloud(

background_color = 'white',

stopwords = set(STOPWORDS),

max_words = 100,

max_font_size = 40,

scale = 5,

random_state = 1

).generate(str(data_copy[data.sentiment == 1].review))
```

```
fig = plt.figure(1, figsize=(20,20))
```

```
plt.axis('off')

plt.imshow(word_cloud)

plt.title('Word cloud for positive opinions', fontsize = 20)

fig.subplots_adjust(top=2.3)

plt.show()
```

#Word cloud for positive opinions

```
word_cloud = WordCloud(

background_color = 'white',
```

```
stopwords = set(STOPWORDS),  
max_words = 100,  
max_font_size = 40,  
scale = 5,  
random_state = 1  
).generate(str(data_copy[data.sentiment == 0].review))
```

```
fig = plt.figure(1, figsize=(20,20))  
plt.axis('off')  
plt.imshow(word_cloud)  
plt.title('Word cloud for negative opinions', fontsize = 20)
```

```
fig.subplots_adjust(top=2.3)
```

```
plt.show()
```

```
def get_corpus(text):
```

```
    words = []
```

```
    for i in text:
```

```
        for j in i.split():
```

```
            words.append(j.strip())
```

```
    return words
```

```
corpus = get_corpus(data.review)
```

```
corpus[:5]
```

```
from collections import Counter  
counter = Counter(corpus)  
most_common = counter.most_common(10)
```

```
most_common = pd.DataFrame(most_common,columns = ['corpus','countv'])

most_common

most_common = most_common.sort_values('countv')
```

Machine Learning

#splitting into train and test

```
train, test= train_test_split(data, test_size=0.2, random_state=42)

Xtrain, ytrain = train['review'], train['sentiment']

Xtest, ytest = test['review'], test['sentiment']
```

TF-IDF and Count Vectorizer

#Vectorizing data

```
tfidf_vect = TfidfVectorizer() #tfidfVectorizer

Xtrain_tfidf = tfidf_vect.fit_transform(Xtrain)

Xtest_tfidf = tfidf_vect.transform(Xtest)
```

```
count_vect = CountVectorizer() # CountVectorizer

Xtrain_count = count_vect.fit_transform(Xtrain)

Xtest_count = count_vect.transform(Xtest)
```

Multinomial Naïve Bayes

```
mnb= MultinomialNB()

mnb.fit(Xtrain_tfidf,ytrain)
```

```

p2=mnb.predict(Xtest_tfidf)

mnb_report = classification_report(ytest, p2, output_dict=True)

prediction_reports = []

prediction_reports.append({
    "Model": "Multinomial Naives Bayes", "precision": mnb_report['weighted
avg']['precision'],
    "Recall": mnb_report['weighted avg']['recall'], "F-score": mnb_report['weighted avg']['f1-
score'],
    "accuracy": mnb_report['accuracy']
})

s2=accuracy_score(ytest,p2)

print("Multinomial Naive Bayes Classifier Accuracy :", "{:.2f}%".format(100*s2))

plot_confusion_matrix(mnb, Xtest_tfidf, ytest,cmap = 'Blues')

plt.grid(False)

```

```

classification_metrics = classification_report(ytest, p2)

print(classification_metrics)

```

Deep Learning Model –LSTM and Transformers

```

def plotCurve(history,epochs):
    epochRange = range(1,epochs+1)

    fig , x = plt.subplots(1,2,figsize = (10,5))

```

```
x[0].plot(epochRange,history.history['accuracy'],label = 'Training Accuracy')
```

```

x[0].plot(epochRange,history.history['val_accuracy'],label = 'Validation Accuracy')

x[0].set_title("Training and Validation accuracy")

x[0].set_xlabel('Epoch')

x[0].set_ylabel('Accuracy')

x[0].legend()

x[1].plot(epochRange,history.history['loss'],label = 'Training Loss')

x[1].plot(epochRange,history.history['val_loss'],label = 'Validation Loss')

x[1].set_title('Training and Validation loss')

x[1].set_xlabel('Epoch')

x[1].set_ylabel('Loss')

x[1].legend()

fig.tight_layout()

plt.show()

```

#splitting into train and test

```

data_copy['review']=data_copy['review'].apply(preprocess_text,lemmatize = False)

#converting target variable to numerical value

data_copy.sentiment = [ 1 if each == "positive" else 0 for each in data_copy.sentiment]

train, test= train_test_split(data_copy, test_size=0.2, random_state=42)

Xtrain, ytrain = train['review'], train['sentiment']

Xtest, ytest = test['review'], test['sentiment']

```

LSTM

#set up the tokenizer

```

MAX_VOCAB_SIZE = 10000

```

```
tokenizer = Tokenizer(num_words = MAX_VOCAB_SIZE,oov_token("<oov>"))

tokenizer.fit_on_texts(Xtrain)

word_index = tokenizer.word_index
```

V = len(word_index)

```
print("Vocabulary of the dataset is : ",V)
```

#generate sequences

```
seq_train = tokenizer.texts_to_sequences(Xtrain)

seq_test = tokenizer.texts_to_sequences(Xtest)
```

maximum sequence length

```
seq_len_list = [len(i) for i in seq_train + seq_test]
```

#if we take the direct maximum then

```
max_len=max(seq_len_list)

print('Maximum length of sequence in the list: {}'.format(max_len))
```

when setting the maximum length of sequence, variability around the average is used.

```
max_seq_len = np.mean(seq_len_list) + 2 * np.std(seq_len_list)

max_seq_len = int(max_seq_len)

print('Maximum sequence length after accounting for standard deviations from average:

{}'.format(max_seq_len))
```

```
perc_data_covered = np.sum(np.array(seq_len_list) < max_seq_len) / len(seq_len_list)*100  
print('The above calculated number coveres approximately {} % of  
data'.format(np.round(perc_data_covered,2)))
```

#create padded sequences

```
pad_train=pad_sequences(seq_train,truncating = 'post', padding = 'pre',maxlen=max_seq_len)  
pad_test=pad_sequences(seq_test,truncating = 'post', padding = 'pre',maxlen=max_seq_len)
```

#Splitting training set for validation purposes

```
Xtrain,Xval,ytrain,yval=train_test_split(pad_train,ytrain,  
                                         test_size=0.2,random_state=10)
```

```
def lstm_model(Xtrain,Xval,ytrain,yval,V,D,maxlen,epochs):
```

```
    print("----Building LSTM Model----")  
    i = Input(shape=(maxlen,))  
    x = Embedding(V + 1, D,input_length = maxlen)(i)  
    x = BatchNormalization()(x)  
    x = Dropout(0.3)(x)  
    x = Conv1D(32,5,activation = 'relu')(x)  
    x = Dropout(0.3)(x)  
    x = MaxPooling1D(2)(x)  
    x = Bidirectional(LSTM(128,return_sequences=True))(x)  
    x = LSTM(64)(x)  
    x = Dropout(0.5)(x)
```

```

x = Dense(1, activation='sigmoid')(x)

model = Model(i, x)

model.summary()

#Training the LSTM

print("---Training LSTM Model---")

model.compile(optimizer= Adam(0.0005),

              loss='binary_crossentropy',

              metrics=['accuracy'])




r = model.fit(Xtrain,ytrain,

               validation_data = (Xval,yval),

               epochs = epochs,

               verbose = 2,

               batch_size = 32)

#callbacks = callbacks

print("Train score:", model.evaluate(Xtrain,ytrain))

print("Validation score:", model.evaluate(Xval,yval))

number_epochs = len(r.history['loss'])




return r,model,number_epochs

D = 64 #embedding dims

epochs = 5

r,model,number_epochs = lstm_model(Xtrain,Xval,ytrain,yval,V,D,max_seq_len,epochs)

```

```
#Plot accuracy and loss  
plotCurve(r,number_epochs)  
  
print("Evaluate Model Performance on Test set")  
result = model.evaluate(pad_test,ytest)  
print(dict(zip(model.metrics_names, result)))
```

```
#Generate predictions for the test dataset  
ypred_label = model.predict(pad_test)  
ypred_label = ypred_label>0.5  
  
#Get the confusion matrix  
conf_mat = confusion_matrix(ytest, ypred_label)  
sns.heatmap(conf_mat, annot = True, fmt ='g', cmap='Blues')  
plt.xlabel('Predicted label')  
plt.ylabel('True label')  
plt.show()
```

```
classification_metrics = classification_report(ytest, ypred_label)  
print(classification_metrics)
```

BERT

```
train, test= train_test_split(data_copy, test_size=0.2, random_state=42)  
Xtrain, ytrain = train['review'], train['sentiment']  
Xtest, ytest = test['review'], test['sentiment']
```

#train and validation sets

```
Xtrain,Xval,ytrain,yval=train_test_split(Xtrain,ytrain,  
                                         test_size=0.2,random_state=10)
```

#tokenization

```
tokenizer = transformers.AutoTokenizer.from_pretrained('distilbert-base-uncased')
```

#pass our texts to the tokenizer.

```
Xtrain_c = tokenizer(Xtrain.tolist(), max_length=max_seq_len,  
                     truncation=True, padding='max_length',  
                     add_special_tokens=True, return_tensors='np')
```

```
Xval_c = tokenizer(Xval.tolist(), max_length=max_seq_len,  
                    truncation=True, padding='max_length',  
                    add_special_tokens=True, return_tensors='np')
```

```
Xtest_c = tokenizer(Xtest.tolist(), max_length=max_seq_len,  
                     truncation=True, padding='max_length',  
                     add_special_tokens=True, return_tensors='np')
```

#preparing our datasets

```
train_dataset = tf.data.Dataset.from_tensor_slices((  
    dict(Xtrain_c),  
    ytrain  
)  
  
val_dataset = tf.data.Dataset.from_tensor_slices((  
    dict(Xval_c),  
    yval
```

```

))
test_dataset = tf.data.Dataset.from_tensor_slices((
    dict(Xtest_c),
    ytest
))

def transformer_model(train_dataset,val_dataset,transformer,max_len,epochs):
    print("---Building transformer model---")
    input_ids = Input(shape=(max_len,), dtype=tf.int32, name="input_ids")
    attention_mask = Input(shape=(max_len,), dtype=tf.int32, name='attention_mask')
    #attention mask
    sequence_output = transformer(input_ids,attention_mask)[0]
    cls_token = sequence_output[:, 0, :]
    x = Dense(512, activation='relu')(cls_token)
    x = Dropout(0.1)(x)
    y = Dense(1, activation='sigmoid')(x)
    model = Model(inputs=[input_ids,attention_mask], outputs=y)
    model.summary()
    model.compile(Adam(lr=2e-5), loss='binary_crossentropy', metrics=['accuracy'])
    r = model.fit(train_dataset.batch(32),batch_size = 32,
                  validation_data = val_dataset.batch(32),epochs = epochs)

    print("Train score:", model.evaluate(train_dataset.batch(32)))
    print("Validation score:", model.evaluate(val_dataset.batch(32)))
    number_epochs = len(r.history['loss'])

```

```

return r,model,number_epochs

transformer = transformers.TFDistilBertModel.from_pretrained('distilbert-base-uncased')

epochs = 2

max_len = max_seq_len

r,model,number_epochs =

transformer_model(train_dataset,val_dataset,transformer,max_len,epochs)

#Plot accuracy and loss

plotCurve(r,number_epochs)

print("Evaluate Model Performance on Test set")

result = model.evaluate(test_dataset.batch(32))

print(dict(zip(model.metrics_names, result)))

#Generate predictions for the test dataset

TM = model.predict(test_dataset.batch(32))

TM = TM>0.5

#Get the confusion matrix

confusion_mat = confusion_matrix(ytest, TM)

sns.heatmap(confusion_mat,annot = True,fmt ='d')

```

```

plt.xlabel('Predicted values')

plt.ylabel('Real values')

plt.show()

classification_metrics = classification_report(ytest, TM)

print(classification_metrics)

```

Appendix E Details analysis of collected dataset

Table 1 Users mentioned most often within the digital media dataset.

Username	User	Times Mentioned
@OpenAI	OpenAI	15394
@elonmusk	Elon Musk	8727
@ChatGPT	ChatGPT	4075
@YouTube	YouTube	3107
@Microsoft	Microsoft	2838
@Google	Google	2543
@sama	Sam Altman	2010
@bing	Bing	1348
@BetaMoroney	Tony Moroney	885
@SpirosMargaris	Spiros Margaris	882

Table 2: Hashtags used most often within the dataset

Hashtag	Times Used
#ChatGPT	283901
#AI	80181
#chatgpt	54299
#ai	24776
#OpenAI	19996
#ArtificialIntelligence	17545
#chatGPT	14295
#openai	13031
#artificialintelligence	8213
#technology	7802