Sanjivani Rural Education Society’s

Sanjivani College of Engineering, Kopargaon, Dist. Ahmednagar (423 603)

(An Autonomous Institute)

**DEPARTMENT OF COMPUTER ENGINEERING**



A

Mini Project Report on

**SOCIAL MEDIA SENTIMENT ANALYSIS**

**Submitted By**

Deokar Om Rajendra (38)

Gaikwad Tushar Raju (47)

Gawali Kaushal Ramesh (50)

Ghanghav Piyush Dyaneshwar (51)

***In the partial Fulfillment for the award of the degree of***

***B.Tech in Computer Engineering***

**Prof. S.A. Shivarkar Dr. D.B. Kshirsagar**

**(Course I/C) (H.O.D.)**

|  |  |  |
| --- | --- | --- |
|  | **Sanjivani Rural Educational Society’s**  **SANJIVANI COLLEGE OF ENGINEERING**  **(An Autonomous Institution)**  Kopargaon – 423 603, Maharashtra. | **ACAD-F-15 K** |
| **Academic Year:** 2023-24 | **CIA ACTIVITY REPORT** | **Revision : 00**  **Dated : \_\_\_\_\_\_\_\_** |
| **Department :** | **Computer Engineering Department** | **Date of Preparation : 28.04.2024** |
| **Course Code & Name:** | **Data Mining and Warehousing (CO314)** | **Year/Sem: T.Y.-A / II** |

**Activity Title**: **Project Based Learning**

**Title: Social Media Sentiment Analysis**

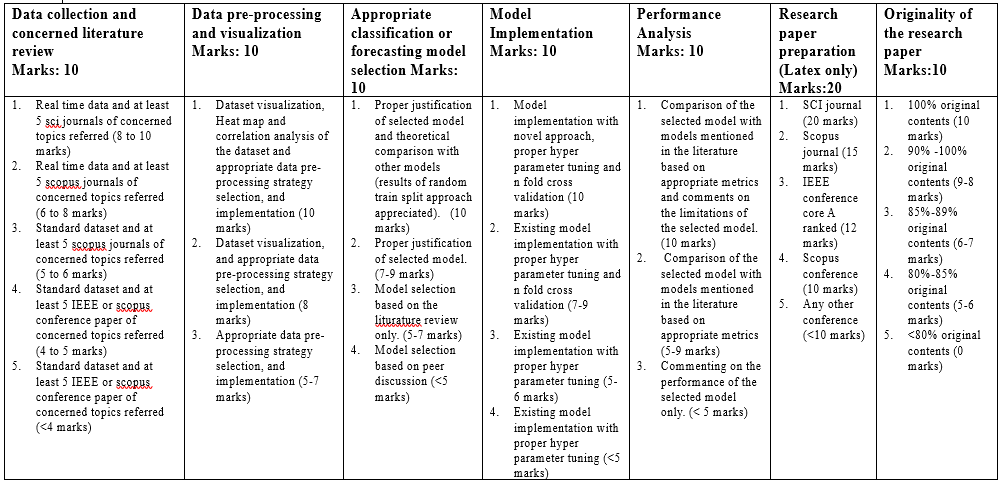
**Submitted by:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll No** | **PRN No** | **Name of the Student** | **Signature** |
| 38 | UCS21M1038 | Deokar Om Rajendra |  |
| 47 | UCS21M1047 | Gaikwad Tushar Raju |  |
| 50 | UCS21M1050 | Gawali Kaushal Ramesh |  |
| 51 | UCS21M1051 | Ghanghav Piyush Dyaneshwar |  |

Prof. S. A. Shivarkar

(Course In-charge)

**Rubrics for Project Based Learning: (20 Marks)**

****

**Project Based Learning (20 Marks)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Roll No of the Students** | **Data collection and concerned literature review**  **Marks: 10** | **Data pre-processing and visualization Marks: 10** | **Appropriate classification or forecasting model selection Marks: 10** | **Model Implementation**  **Marks: 10** | **Performance Analysis**  **Marks: 10** | **Research paper preparation**  **(Latex only)**  **Marks:20** | **Originality of the research paper**  **Marks:10** |
| 38 |  |  |  |  |  |  |  |
| 47 |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |
| 51 |  |  |  |  |  |  |  |

**Acknowledgement**

First and foremost, I express my deep sense of gratitude, sincere thanks to **Prof. S. A. Shivarkar** and **Dr. Ajit Muzumdar** Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon. Your availability at any time throughout the semester, encouragement and support tremendously boosted this project work. Lots of thanks to Head of Computer Engineering Department, **Dr. D.B. Kshirsagar** for providing us the best support. I would like to express my sincere gratitude to **Dr. A.G. Thakur,** Director, Sanjivani College of Engineering, Kopargaon for providing great platform to complete the project within the scheduled time.

Deokar Om Rajendra(38)

Gaikwad Tushar Raju(47)

Gawali Kaushal Ramesh(50)

Ghanghav Piyush Dyaneshwar (51)

**Abstract**

Every Day by day the social network development of communication and the data on the websites is increasing very rapidly. ”The social media systems have advanced rapidly and many individuals regularly utilize these administrations to interact with others and designate themselves by sharing their conclusions, sees and way of thinking. It is more often than not apportioned by winnowing the corresponding or interlinked occasions specified on social media websites like Telegram, YouTube, YouTube and Facebook etc. A social media sentiment analysis has rooted from being a unidirectional to a bidirectional dialogue i.e. between eras to era. The elementary objective behind such examinations is to imagine the degree of criticality with importance feedback or appreciation illustrate at intervals the comments, re-tweets or blogs. This investigation appears how individuals expect, survey, interaction and opinion concerning completely different problems. This paper centers on the design area and look at the feelings interacted in a specific sentence, entry or archive. The suppositions and opinions are release from the collected information and it is feasibly assessed based on the degree of quality. The run the show commitment of this paper is to deliver a blueprint to the people who attempt to utilize web based life scratching and examination utilizing distinctive programming equipment either in their trade. In expansion, the system design to execute the proposed thought has been discussed. Index terms sScraping, NLP, textBlob, estimation analysis, trend investigation.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Title** | **Page No** |
| 1. | Introduction | 7 |
| 2. | Scope, Objectives and Challenges | 8-9 |
| 3. | Literature Review | 10-11 |
| 4. | System Requirements Specifications | 12-15 |
| 5. | System Architecture | 16-17 |
| 6. | Methodologies | 18-20 |
| 7. | Implementation Details | 21-28 |
| 8. | Result Analysis and Discussion | 29 |
| 9. | Conclusions | 30 |
| 10. | References | 30 |

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Name of the Figure** | **Page No** |
| 5.1 | System Architecture | 16 |
| 7.1 | Normalized Data creating new CSV file | 22 |
| 7.2 | Labelled encoded data to calculate Correlation | 23 |
| 7.3 | Graphic User Interface (GUI) | 28 |
| 8.1 | Support Vector Machine Performance analysis | 29 |
| 8.2 | Random Forest Performance analysis | 29 |
| 8.3 | Decision Tree Performance analysis | 29 |
| 8.4 | Correlation Heatmap | 29 |
| 8.5 | Naïve Bayes Accuracy | 29 |

**Chapter 1**

**Introduction**

The social media networks have evolved quickly and many people often use these services to communicate with others and categorize themselves by sharing their opinions, views and concepts. It is usually dispensed by winnowing the corresponding or interlinked events mentioned on social media websites like twitter, Facebook, YouTube and Pinterest etc. Despite the advancement in the email filtering techniques, the consistent barrage of spam emails continues to annoy the users, now emerging as a endless problem, it results into in filtering inboxes with unwanted and often resentful content. This not only inundates users with irrelevant and certainly hazardous content as well as poses serious security risks such as phishing scams and malware distribution.

This paper centers on the pattern location and examine the feelings communicated in a particular sentence, passage or archive. We know that nowadays, social media has become an important part of our lives. We use social media for communication, interaction, sharing knowledge, etc. With billions of active users worldwide, platforms such as Facebook, Twitter, Instagram, and LinkedIn have transformed into powerful channels for expressing opinions, discussing current events, and shaping public discourse. One area where social media has played a significant role is in raising awareness and mobilizing support for global issues, including the sustainable development goals (SDGs). The SDGs, adopted by the United Nations in 2015, encompass a set of 17 interconnected goals aimed at addressing the world's most pressing social, economic, and environmental challenges. Achieving the SDGs requires collective action from governments, businesses, civil society, and individuals, making public awareness and engagement crucial for their success.

We see that china and india have the huge users in both years. China and india have a population of over 140.76 crores, and because of this, india and china have the most users of social media. Social media platforms have provided a dynamic space for individuals, organizations, and movements to advocate for the SDGs, share success stories, and call for action. Moreover, social media offers a vast repository of user-generated content that can be analyzed to gauge public sentiment and track evolving trends regarding sustainable development. By analyzing social media data, researchers, policymakers, and organizations can gain valuable insights into the public's perception, engagement, and sentiment towards the SDGs, enabling them to refine strategies, target interventions, and measure progress.

**Chapter 2**

**Scope, Objectives and Challenges**

**Scope:**

1. **Data Collection:** Gathering data from various social media platforms, such as Twitter, Facebook, and Instagram, using APIs or web scraping techniques. This involves retrieving a diverse range of content, including posts, comments, and reviews, related to specific topics or keywords of interest.
2. **Text Preprocessing:** Cleaning and preprocessing the collected data to remove noise, such as special characters, emojis, and irrelevant information. Techniques like tokenization, stemming, and stop-word removal are applied to prepare the text data for sentiment analysis.
3. **Feature Extraction**: Extracting meaningful features from the preprocessed text data to represent the content in a format suitable for sentiment analysis. This may involve techniques like word embeddings, bag-of-words representations, or other feature engineering methods.
4. **Sentiment Analysis**: Applying machine learning algorithms, such as Support Vector Machines (SVM), Naive Bayes, or deep learning models like Recurrent Neural Networks (RNNs) or Transformers, to classify the sentiment of social media content into positive, negative, or neutral categories.
5. **Model Evaluation**: Evaluating the performance of the sentiment analysis models using appropriate metrics, such as accuracy, precision, recall, and F1-score. This involves testing the models on labeled datasets and fine-tuning parameters to optimize performance.
6. **Data Visualization**: Visualizing sentiment analysis results using data visualization techniques, such as bar charts, pie charts, or word clouds, to present insights effectively. This allows stakeholders to interpret and understand the sentiment distribution across different social media platforms and topics.
7. **Ethical Considerations**: Addressing ethical considerations and privacy regulations to ensure responsible data usage and protection of user privacy throughout the project lifecycle. This includes obtaining consent for data collection, anonymizing sensitive information, and adhering to relevant data protection laws.

**Objectives:**

1. **Data Acquisition:** Gather data from various social media platforms, ensuring a diverse and representative sample related to the topics of interest.
2. **Text Preprocessing**: Clean and preprocess the collected data to remove noise and irrelevant information, making it suitable for sentiment analysis.
3. **Feature Extraction**: Extract meaningful features from the preprocessed text data to represent the content effectively for sentiment analysis.
4. **Model Development**: Develop and train machine learning models, including Support Vector Machines (SVM), Naive Bayes, and deep learning models like Recurrent Neural Networks (RNNs) or Transformers, to classify social media content into positive, negative, or neutral sentiments.
5. **Model Evaluation:** Evaluate the performance of the sentiment analysis models using appropriate metrics and test them on labeled datasets to ensure accuracy and reliability.
6. **Data Visualization**: Visualize sentiment analysis results using data visualization techniques to present insights effectively and facilitate interpretation by stakeholders.
7. **Ethical Compliance**: Ensure compliance with ethical considerations and privacy regulations throughout the project lifecycle, including obtaining consent for data collection and protecting user privacy.
8. **Scalability and Efficiency:** Design the system to be scalable and efficient, capable of handling large volumes of social media data and performing sentiment analysis in real-time or near real-time.
9. **Stakeholder Engagement:** Collaborate with domain experts and stakeholders to understand requirements, tailor the analysis to specific use cases or industries, and incorporate feedback into the project.
10. **Documentation and Communication**: Document the project methodology, findings, and results comprehensively, and communicate them effectively to stakeholders through reports, presentations, and demonstrations.

**Challenges:**

1. **Data Quality:** Social media data can be noisy, unstructured, and prone to spam, sarcasm, slang, and misspellings, making it challenging to extract accurate sentiment.
2. **Data Privacy and Ethics**: Ensuring compliance with data privacy regulations and ethical considerations, such as obtaining consent for data collection and protecting user anonymity.
3. **Language and Context**: Social media content often includes multiple languages, dialects, and cultural nuances, requiring robust language processing and context understanding for accurate sentiment analysis.
4. **Scalability:** Analyzing large volumes of social media data in real-time or near real-time requires scalable and efficient algorithms and infrastructure to handle the computational load.
5. **Class Imbalance**: Imbalanced datasets, where one sentiment class is more prevalent than others, can bias model training and evaluation, requiring techniques to address class imbalance.
6. **Domain Specificity**: Social media sentiment analysis may require domain-specific knowledge and customization to accurately interpret sentiment in specific industries or topics.
7. **Model Overfitting**: Machine learning models may overfit to training data, capturing noise and irrelevant patterns, leading to poor generalization performance on unseen data.
8. **Data Labeling:** Manual labeling of social media data for training sentiment analysis models can be time-consuming, expensive, and subjective, requiring efficient labeling strategies and quality control measures.
9. **Interpretability**: Interpreting and explaining the decisions made by sentiment analysis models, especially deep learning models, can be challenging due to their black-box nature, limiting trust and transparency.

**Chapter 3**

**Literature Review**

**1]** **Research Paper Name: Sentimental Analysis of Twitter Comments on Covid-19**

**Author: Supriya Raheja and Anjani Asthana**

Observation : Some of the studies reported the psychological effect of lockdown on human behavior like anxiety, depression,frustration etc. and this was reflected while using social media, their comments reflected their behavior. In the paper, we have seen the textual analyses of Twitter information to know the public views tracking the advancement of fear, which is directly related with the vast spread of Coronavirus disease. The key focus of the Sentimental Analysis is on textual data which requires more text processing. To represent how many times the word had come “COVID” in a twitter dataset,a WordCloud is made. This paper performed the sentimental analysis for three important keywords (COVID, CORONA VIRUS, COVID – 19) of COVID-19.

**2]** **Research Paper Name: Sentimental Analysis using Capsule Network with Gravitational Search Algorithm**

**Author: V. Diviya Prabha and R. Rathipriya**

Observation : This work is to propose an algorithm which is combination of Capsule Network (CN) with Gravitational Search Algorithm (GSA) to analyze people’s sentiments from twitter data. The information available in tweets post are unstructured format and generally difficult to analyze. Algorithm like Gravitational Search Algorithm (GSA) is used to search space in a minimal architecture. It works on the concept of force and mass every words in the text attracts the other related words here particle act as agents. The proposed model overcomes different data pre-processing techniques and makes it suitable for converting the words to vectors encoding.

**3] Research Paper Name: Performance of Sentimental Analysis by Studying and Mining Social Media using Parsing Technique**

**Author: Ms C. M. Sowmya, Ms. S. Deena and Dr. S. Anbuchelian.**

Observation : One purpose behind pattern examination might be to find an early or suspicious conduct occurring on the online life stage. The level of pattern will furthermore be grouped into low, medium, high which can basically be associated with the fundamental remarks or tweets. The Python dialect can be utilized for normal dialect discovery, title and substance extraction, question coordinating and, once used in conjunction with a module like scikit-learn, it might be prepared to perform assessment examination abuse Naïve Bayes classifier.

**4]** **Research Paper Name: Analyzing Sentimental Influence of Posts on Social Networks**

**Author: Beiming Sun and Vincent TY Ng.**

Observation : One purpose behind pattern examination might be to find an early or suspicious conduct occurring on the online life stage. The level of pattern will furthermore be grouped into low, medium, high which can basically be associated with the fundamental remarks or tweets. The Python dialect can be utilized for normal dialect discovery, title and substance extraction, question coordinating and, once used in conjunction with a module like scikit-learn, it might be prepared to perform assessment examination abuse Naïve Bayes classifier.

**5]** **Research Paper Name: A systematic review of social media-based sentiment analysis: Emerging trends and challenges**

**Author: Qianwen Ariel Xu , Victor Chang and Chrisina Jayne**

Observation :Massive amounts of user-generated data sourced from various social networking platforms also provide new insights for businesses and governments.Sentiment analysis provides an automated method of analyzing sentiment, emotion and opinion.The volume of data continues to increase, how to extract useful information effectively and efficiently from the vast amount of data has become a critical issue.A large number of scholars have worked on improving the performance of various sentiment classifiers or applying them to various domains using data from social networking platforms.The paper identifies the challenges and issues in the existing sentiment analysis research.

**Chapter 4**

**System Requirement Specifications**

**• Normal Requirements:**

1. Data Collection:
   * Define the social media platforms to be analyzed (e.g., Twitter, Facebook, Reddit).
   * Specify the timeframe for data collection.
   * Determine the criteria for selecting relevant data (e.g., hashtags, keywords, user demographics).
2. Preprocessing:
   * Text cleaning: Remove noise such as URLs, special characters, and emojis.
   * Tokenization: Split text into individual words or tokens.
   * Stopword removal: Eliminate common words that do not contribute to sentiment analysis.
   * Lemmatization or stemming: Normalize words to their base forms to reduce dimensionality.
3. Sentiment Analysis Model:
   * Select an appropriate sentiment analysis model (e.g., lexicon-based, machine learning, deep learning).
   * Train the model using labeled data for supervised learning approaches.
   * Fine-tune the model parameters to optimize performance.
4. Evaluation:
   * Define evaluation metrics (e.g., accuracy, precision, recall, F1-score).
   * Split the dataset into training, validation, and test sets.
   * Evaluate the model's performance using the selected metrics on the test set.
   * Compare the results with baseline models or existing sentiment analysis tools.
5. Interpretation:
   * Analyze the sentiment distribution across different social media platforms.
   * Identify key themes or topics driving positive or negative sentiment.
   * Explore correlations between sentiment and external factors (e.g., events, news, user demographics).
6. Ethical Considerations:
   * Ensure compliance with data privacy regulations (e.g., GDPR, CCPA).
   * Address potential biases in the dataset or model predictions.
   * Transparency in reporting findings and limitations of the analysis.
7. Deployment:
   * Develop user-friendly interfaces for accessing sentiment analysis results.
   * Integrate sentiment analysis into existing applications or platforms.
   * Provide documentation and support for users deploying the sentiment analysis model.
8. Maintenance and Updates:
   * Monitor model performance over time and retrain as necessary.
   * Update sentiment lexicons or algorithms to adapt to evolving language and social trends.
   * Incorporate user feedback to improve the accuracy and relevance of sentiment analysis results.

**• Expected Requirements:**

1. Scalable Data Processing:
   * Ability to handle large volumes of social media data efficiently.
   * Utilization of distributed computing or cloud resources for scalability.
2. Real-time Data Ingestion:
   * Capability to ingest streaming data from social media platforms in real-time.
   * Integration with APIs or streaming data pipelines for continuous data collection.
3. Multilingual Support:
   * Support for analyzing sentiment in multiple languages to cater to diverse user bases.
   * Integration of language detection and translation capabilities where necessary.
4. Domain-specific Lexicons:
   * Creation or adoption of domain-specific sentiment lexicons tailored to the topics of interest.
   * Regular updates to lexicons to incorporate domain-specific language nuances.
5. Contextual Understanding:
   * Incorporation of context-aware sentiment analysis techniques to account for sarcasm, irony, and colloquial expressions.
   * Use of contextual embeddings or transformer models for capturing nuanced meanings.
6. Multimodal Analysis:
   * Integration of text, image, and video analysis for comprehensive sentiment understanding.
   * Fusion of information from different modalities to enhance sentiment prediction accuracy.
7. Interactive Visualization:
   * Development of interactive dashboards or visualizations for exploring sentiment trends.
   * Support for drill-down capabilities to investigate sentiment at different levels of granularity.
8. Automated Insights Generation:
   * Implementation of algorithms for automatically summarizing key insights from sentiment analysis results.
   * Generation of actionable recommendations based on sentiment trends and patterns.
9. Robustness and Reliability:
   * Implementation of error handling mechanisms to ensure robustness against data inconsistencies or failures.
   * Adoption of fault-tolerant architectures to minimize downtime and ensure continuous operation.
10. Compliance and Security:
    * Adherence to data privacy regulations and best practices for handling sensitive user information.
    * Implementation of access controls and encryption techniques to protect data integrity and confidentiality.

**• Excited Requirements:**

1. Real-time Data Streaming:
   * Implement real-time data streaming capabilities to continuously ingest social media data from various platforms such as Twitter, Facebook, Instagram, and LinkedIn.
   * Utilize APIs or web scraping techniques to collect streaming data with minimal latency and ensure up-to-date analysis.
2. Scalability and Elasticity:
   * Design the sentiment analysis system to scale dynamically based on fluctuating data volumes and processing demands.
   * Utilize cloud-based infrastructure or containerization technologies to achieve scalability and elasticity, enabling the system to handle increasing workloads seamlessly.
3. Multilingual Support:
   * Incorporate multilingual support to analyze sentiment across diverse languages and regions.
   * Implement language detection mechanisms to automatically identify the language of social media posts and apply language-specific sentiment analysis models.
4. Advanced NLP Techniques:
   * Employ advanced natural language processing (NLP) techniques such as named entity recognition, aspect-based sentiment analysis, and emotion detection.
   * Enhance sentiment analysis algorithms with deep learning models, transformer architectures, and pre-trained language models (e.g., BERT, GPT) for improved accuracy and granularity.
5. Customizable Sentiment Lexicons:
   * Provide the ability to customize sentiment lexicons and dictionaries to adapt to specific domains, industries, or user preferences.
   * Allow users to define custom sentiment rules, keywords, and thresholds to tailor sentiment analysis to their unique requirements.
6. Interactive Visualization Tools:
   * Develop interactive visualization tools and dashboards to present sentiment analysis results in an intuitive and actionable format.
   * Enable users to explore sentiment trends, sentiment distributions, and sentiment correlations across different social media platforms and time periods.
7. Sentiment-driven Alerts and Notifications:
   * Implement sentiment-driven alerting mechanisms to notify users of significant changes or anomalies in social media sentiment.
   * Define customizable alert thresholds based on sentiment polarity, sentiment intensity, or specific keywords to trigger timely notifications.
8. Integration with Social Media Management Tools:
   * Integrate with popular social media management tools and platforms (e.g., Hootsuite, Sprout Social) to streamline data collection, analysis, and reporting workflows.
   * Enable seamless data exchange and collaboration between sentiment analysis systems and social media marketing teams.
9. Continuous Model Training and Improvement:
   * Establish mechanisms for continuous model training and improvement using feedback loops, active learning, and model retraining techniques.
   * Incorporate user feedback, annotation data, and domain-specific knowledge to iteratively refine sentiment analysis models and enhance prediction accuracy over time.

**• Hardware Requirements:**

**1. Device name:** Acer Nitro 5

**2. Processor:** AMD Ryzen

**3. RAM:** 8.00 GB

**4. System type:** 64-bit operating system, x64-based processor

**• Software Requirements:**

**1.Operating System:** Windows 11.

**2.Programming Language:** Python.

**3.Python Libraries:**

1. Pandas

2. NumPy

3. Scikit-learn

4. Matplotlib

5. Seaborn

6. tkinter

**4.Integrated Development Environment (IDE):**

1.Visual Studio Code.

2. Spyder

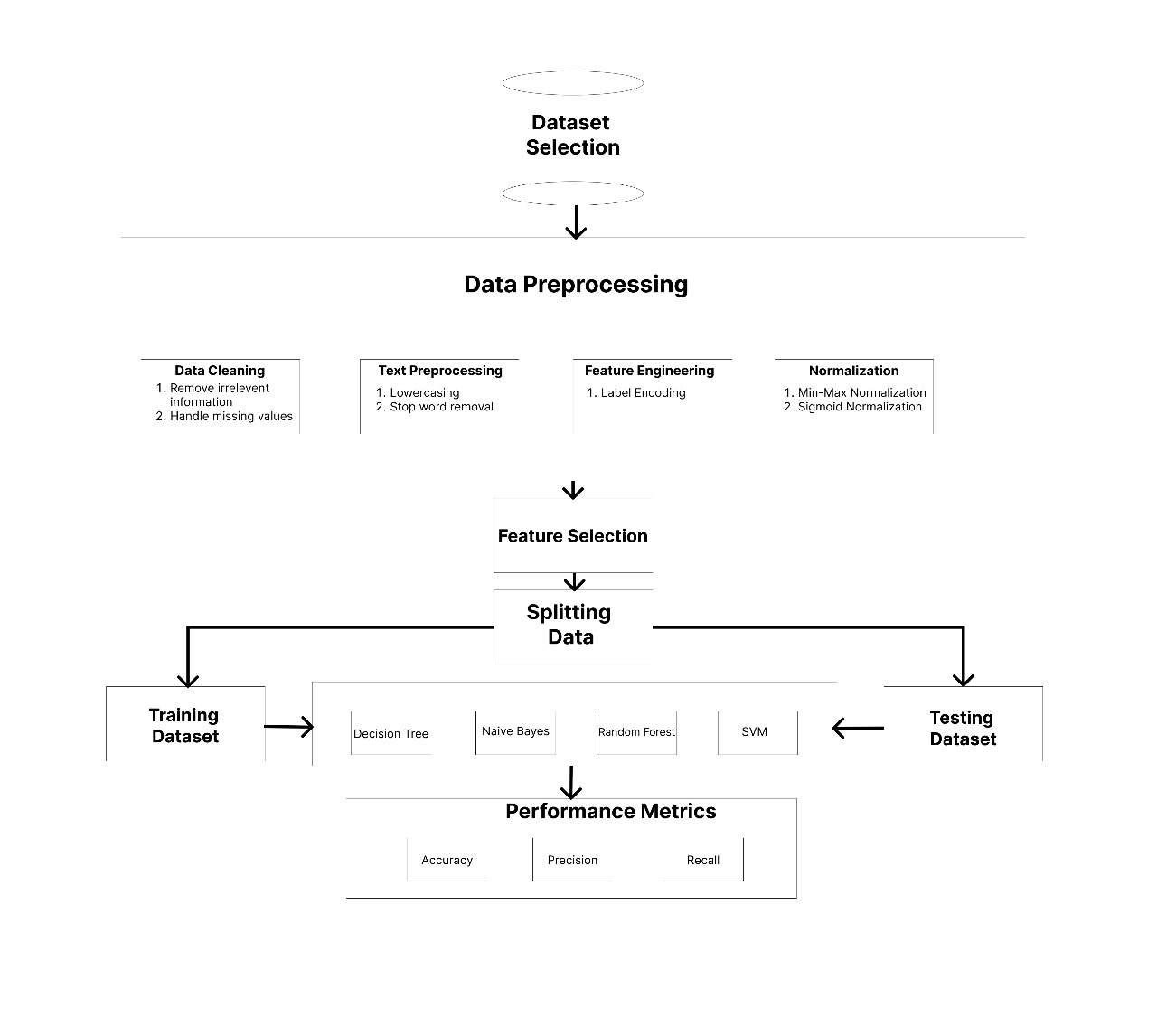
3. Anaconda Navigator.

**5.Version Control:** Git And Github.

**• Data Requirements: Dataset:** https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset

**Chapter 5**

**System Architecture**

**Figure 5.1. System Architecture**

Social media sentiment analysis involves using text processing techniques, machine learning

algorithms, and classification methods to analyze the sentiment expressed in social media

content, such as tweets.

1) Social networking provides a wealth of textual data generated by users expressing

opinions, feelings, and experiences on various topics. This data can be leveraged for

sentiment analysis to understand public perception, customer sentiment, and trends.

2) Text processing techniques are applied to preprocess social media content before

sentiment analysis. This includes removing URLs, filtering repeated words, handling

WH questions, removing special characters, emotions, or emojis, and excluding

retweets to ensure the text is clean and ready for analysis.

3) Pre-processed tweets are then fed into machine learning algorithms. These algorithms

are trained on labelled datasets containing examples of tweets with their corresponding

sentiment labels (positive, negative, or neutral).

4) Machine learning algorithms learn patterns and relationships within the text data to

make predictions about the sentiment of unseen tweets. Common algorithms used for

sentiment analysis include Support Vector Machines (SVM), Naive Bayes, and Neural

Networks.

5) The trained model is then used to classify the sentiment of new tweets into one of three

categories: positive, negative, or neutral. This classification helps to understand the

overall sentiment of social media conversations surrounding a particular topic, brand,

event, or keyword.

6) The output of the sentiment analysis process provides insights into how people feel

about a given topic on social media. By classifying tweets into positive, negative, or

neutral sentiments, businesses, researchers, and organizations can gauge public opinion,

identify emerging trends, monitor brand reputation, and make data-driven decisions to

improve products, services, or communication strategies.

**Chapter 6**

**Methodologies & Module Description**

1. **Module Description:**
2. Label Encoding Module:

* Converts categorical text attributes into numerical labels using label encoding.
* Enables inclusion of textual attributes in quantitative analyses by transforming them into nominal numerical representations.
* Prepares the dataset for correlation analysis and heatmap generation by providing numeric labels for categorical variables.
* Correlation Analysis and Heatmap Generation Module:
* Conducts correlation analysis to examine relationships between variables in the dataset.
* Generates a heatmap visualizing correlation matrices to facilitate interpretation of complex datasets.
* Identifies significant correlations between attributes, aiding in understanding the underlying patterns and dependencies within the data.

1. Min-Max Scaling Module:

* Transforms label-encoded values into continuous values within the range of -1 to +1 using the Min-max scaling technique.
* Preserves relative distances between values while linearly scaling them to a specified range, facilitating comparison and analysis.
* Implements the min-max scaling equation to standardize the data for further processing and modeling.

1. Decision Trees Module:

* Implements decision tree algorithm for classification and regression tasks in machine learning.
* Recursively partitions the feature space into regions associated with specific outcomes or predictions.
* Provides an intuitive and interpretable model for both experts and non-experts, enabling understanding of the decision-making process.

1. Naive Bayes Classifier Module:

* Utilizes the Naive Bayes algorithm for classification tasks based on Bayes' theorem.
* Efficiently handles classification problems, especially in text classification and spam filtering applications.
* Employs the general equation for the Naive Bayes classifier to calculate probabilities and make predictions.

1. Random Forest Module:

* Implements Random Forest, an ensemble learning technique, for classification and regression tasks.
* Constructs multiple decision trees during training and combines their predictions through voting or averaging.
* Known for its robustness and high accuracy across diverse datasets, providing reliable predictions in various scenarios.

1. Support Vector Machines (SVM) Module:

* Implements Support Vector Machines (SVM) for supervised learning tasks, including binary and multi-class classification, and regression.
* Finds the optimal hyperplane that best separates data points belonging to different classes in the feature space.
* Well-suited for classification tasks, offering high accuracy and versatility across different types of datasets.

1. **Methodology:**
2. Data Collection and Preprocessing:

* Gather social media data containing textual attributes relevant to the sentiment analysis task.
* Preprocess the data by removing noise, such as URLs, special characters, and emojis.
* Apply label encoding to convert categorical text attributes into numeric labels, enabling inclusion in quantitative analyses.

1. Correlation Analysis and Heatmap Generation:

* Conduct correlation analysis on the dataset to explore relationships between attributes.
* Generate a heatmap visualizing the correlation matrix to identify significant correlations.
* Analyze the heatmap to gain insights into the interplay between different attributes and their impact on sentiment.

1. Min-max Scaling:

* Convert label-encoded values into continuous values within the range of -1 to +1 using min-max scaling.
* Linearly scale the values while preserving their relative distances, facilitating quantitative analysis and comparison.

1. Decision Trees:

* Utilize decision trees as a classification algorithm to predict sentiment labels based on input features.
* Train decision tree models on the preprocessed dataset, recursively partitioning the feature space.
* Interpret the resulting decision trees to understand the factors influencing sentiment predictions.

1. Naive Bayes Classifier:

* Implement the Naive Bayes classifier for sentiment classification tasks.
* Utilize Bayes' theorem to calculate the probability of sentiment labels given the input features.
* Train the Naive Bayes model on the labeled dataset and evaluate its performance using appropriate metrics.

1. Random Forest:

* Apply Random Forest as an ensemble learning technique for sentiment analysis.
* Construct multiple decision trees during training and combine their predictions through voting.
* Train the Random Forest model on the dataset and assess its accuracy and robustness through cross-validation.

.

1. Support Vector Machines (SVM):

* Employ SVM as a powerful supervised learning model for sentiment classification.
* Train the SVM model to find the optimal hyperplane separating data points of different sentiment classes.
* Evaluate the SVM model's performance on the test dataset and compare it with other classifiers.

1. Evaluation and Interpretation:

* Evaluate the performance of each sentiment analysis model using metrics such as accuracy, precision, recall, and F1-score.
* Interpret the results of correlation analysis, heatmap visualization, and model predictions to gain insights into social media sentiment trends.
* Identify key attributes and features influencing sentiment polarity and strength.

**Chapter 7**

**Implementation Details**

**1.Normalization**

def normalization(self):

# Replace 'your\_file.csv' with the path to your CSV file

input\_filename = 'C:/Users/tusha/OneDrive/Desktop/DMWUI/Heatmap/relevant\_dataset.csv'

output\_filename = 'C:/Users/tusha/OneDrive/Desktop/DMWUI/Normalization/normalized\_dataset.csv'

data = self.read\_csv(input\_filename)

# Extracting columns from the data

sentiment = [float(row[1]) for row in data[1:]] # Assuming sentiment is in the second column (index 1)

retweets = [float(row[3]) for row in data[1:]] # Assuming retweets is in the fourth column (index 3)

likes = [float(row[4]) for row in data[1:]] # Assuming likes is in the fifth column (index 4)

month = [float(row[5]) for row in data[1:]] # Assuming month is in the sixth column (index 5)

hour = [float(row[6]) for row in data[1:]] # Assuming hour is in the seventh column (index 6)

texts = [row[0] for row in data[1:]] # Assuming text is in the first column (index 0)

hashtags = [row[2] for row in data[1:]] # Assuming hashtags is in the third column (index 2)

# Normalizing numerical attributes

normalized\_sentiment = self.normalize(sentiment)

normalized\_retweets = self.normalize(retweets)

normalized\_likes = self.normalize(likes)

normalized\_month = self.normalize(month)

normalized\_hour = self.normalize(hour)

# Normalizing text and hashtags attributes

text\_dict = {text: i for i, text in enumerate(set(texts))}

hashtag\_dict = {hashtag: i for i, hashtag in enumerate(set(hashtags))}

normalized\_texts = self.normalize([text\_dict[text] for text in texts])

normalized\_hashtags = self.normalize([hashtag\_dict[hashtag] for hashtag in hashtags])

# Updating the data with normalized values

updated\_data = []

updated\_data.append(data[0]) # Append header

for i in range(len(data) - 1):

row = [

normalized\_texts[i],

normalized\_sentiment[i],

normalized\_hashtags[i],

normalized\_retweets[i],

normalized\_likes[i],

normalized\_month[i],

normalized\_hour[i]

]

updated\_data.append(row)

# Write the updated data to a new CSV file

self.write\_csv(output\_filename, updated\_data)

messagebox.showinfo("Success", "normalization applied and CSV file generated.")

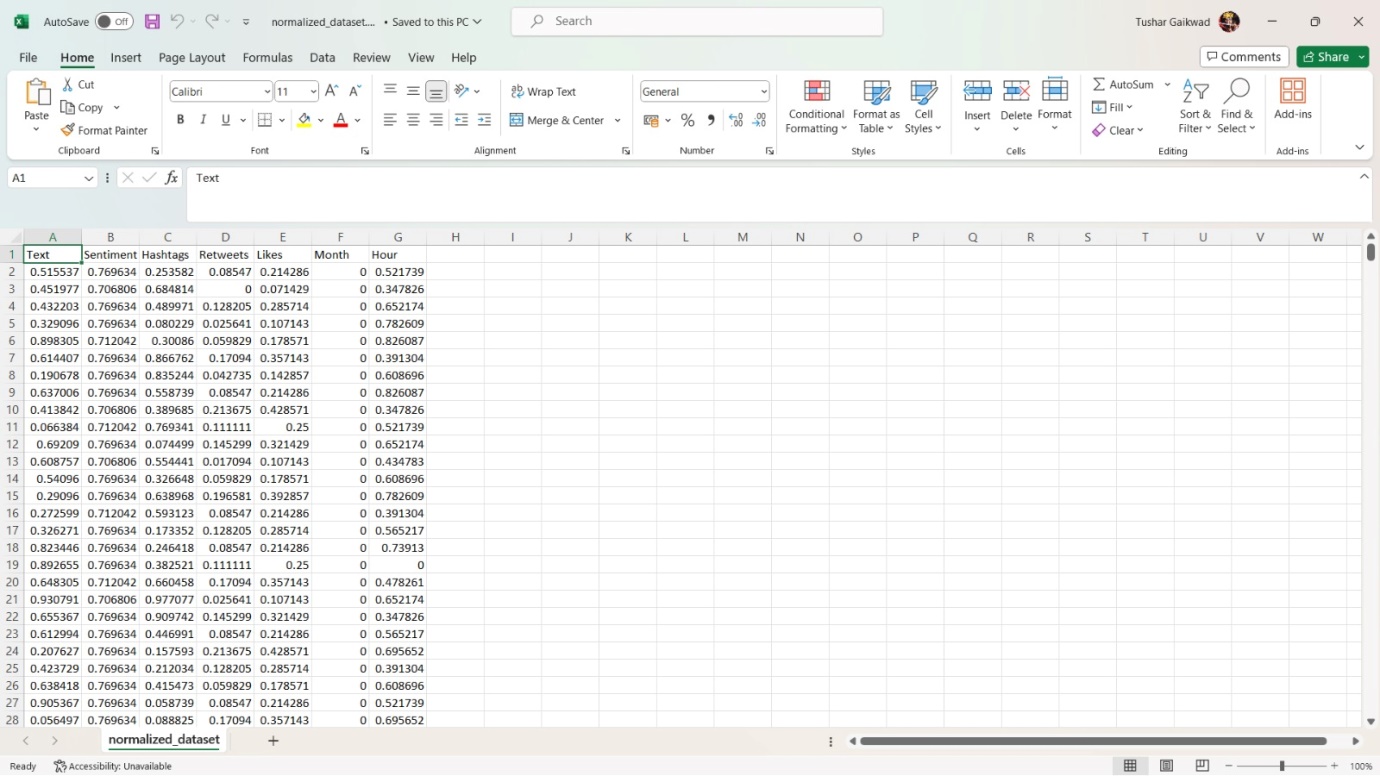


Fig. 7.1: Normalized Data

def normalize(self,attribute):

min\_val = min(attribute)

max\_val = max(attribute)

normalized\_attribute = [(x - min\_val) / (max\_val - min\_val) for x in attribute]

return normalized\_attribute

**2.Label Encoding:**

def label\_encoding(self):

# Load your dataset

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/sentimentdataset.csv')

# Get only the columns with text attributes

text\_columns = df.select\_dtypes(include=['object']).columns

# Initialize LabelEncoder

encoder = LabelEncoder()

# Apply label encoding to each text attribute column

for column in text\_columns:

df[column] = encoder.fit\_transform(df[column])

output\_file\_path = 'C:/Users/tusha/OneDrive/Desktop/DMWUI/Encoded dataset/encoded\_dataset.csv'

# Save the final DataFrame to a CSV file

df.to\_csv(output\_file\_path, index=False)

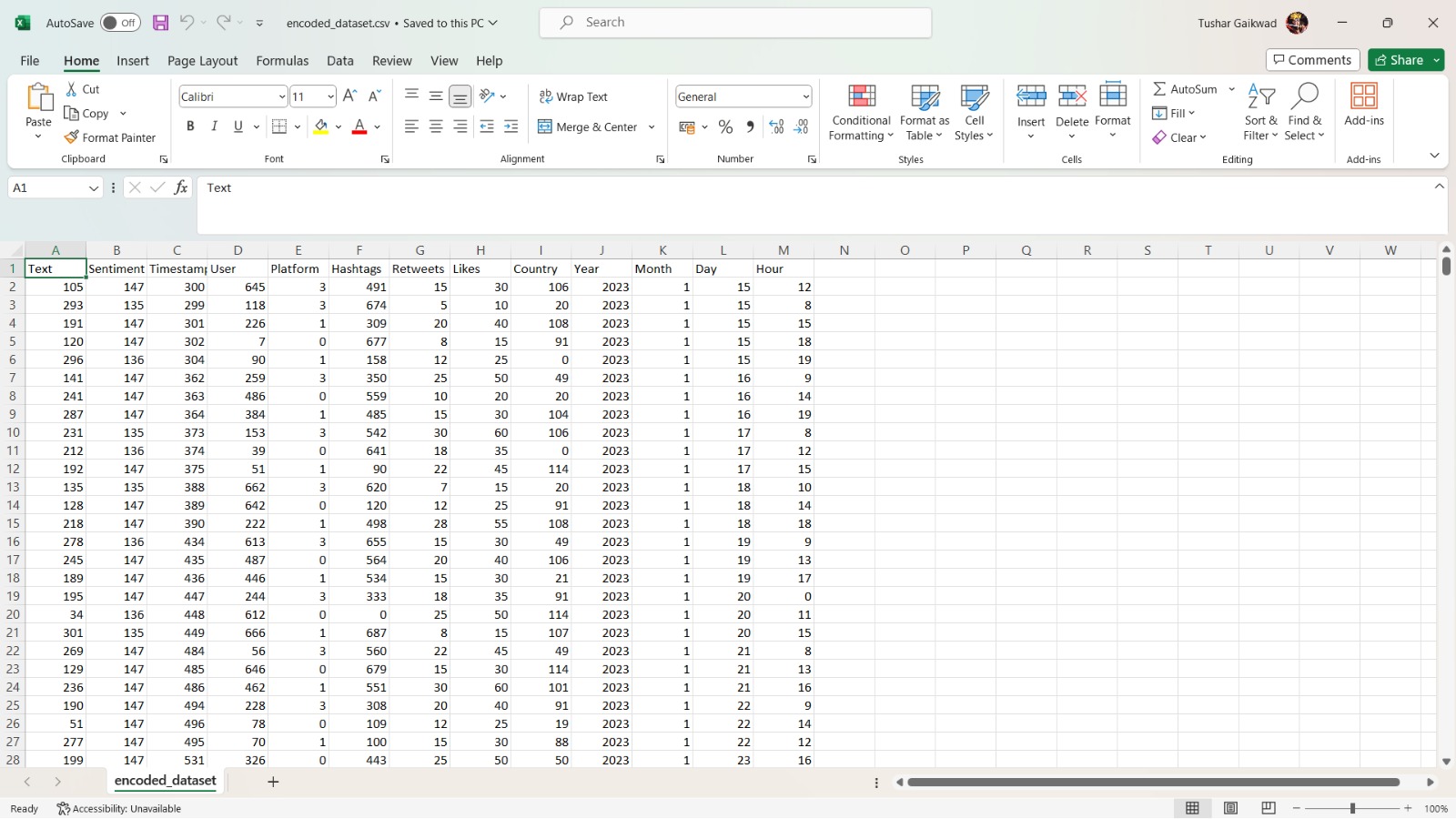


Fig. 7.2: Labelled Encoded Data

**3.Heatmap:**

def heatmap(self):

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Encoded dataset/encoded\_dataset.csv')

# Compute the correlation matrix

corr = df.corr()

corr\_matrix = df.corr(numeric\_only=True)

target\_corr = corr\_matrix['Text']

threshold = 0.1

relevant\_features = target\_corr[abs(target\_corr) > threshold].index.tolist()

X = df[relevant\_features]

# Output path for the relevant dataset

output\_path\_file = "C:/Users/tusha/OneDrive/Desktop/DMWUI/Heatmap/relevant\_dataset.csv"

X.to\_csv(output\_path\_file, index=False)

# Set up the matplotlib figure

plt.figure(figsize=(10, 8))

# Draw the heatmap with the mask and correct aspect ratio

sns.heatmap(corr, annot=True, fmt="0.02f", cmap='coolwarm', linewidths=0.5)

# Add title

plt.title('Correlation Heatmap of 13 Attributes')

# Save the plot as a PDF

plt.savefig('C:/Users/tusha/OneDrive/Desktop/DMWUI/Heatmap/correlation\_heatmap.pdf')

messagebox.showinfo("Success", "Correlation Heatmap and CSV file generated.")

# Show the plot

plt.show()

self.update()

**4.Sigmoid Function:**

def Sigmoide(self):

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Normalization/Normalized\_dataset.csv')

# Apply sigmoid function to all columns except non-numeric ones

# Apply sigmoid function to 'Sentiment' column

df['Sentiment\_Sigmoid'] = self.sigmoid(df['Sentiment'])

# Define the threshold for binary conversion

threshold = 0.664116585874051 # Adjust the threshold as needed

# Apply sigmoid function to all columns except non-numeric ones

df\_numeric = df.select\_dtypes(include=[np.number]) # Select only numeric columns

df[df\_numeric.columns] = df\_numeric.apply(lambda x: x.map(self.sigmoid)) # Apply sigmoid function to numeric columns

# Create a new column with binary labels based on the threshold

df['Sentiment\_Binary'] = (df['Sentiment\_Sigmoid'] >= threshold).astype(int)

# Optionally, display a message box to indicate success

messagebox.showinfo("Success", "Sigmoid function applied and csv file generated.")

# messagebox.showinfo("Success", "Sigmoid function applied and CSV file generated.")

# Define the file path for saving the CSV file

output\_file\_path = 'C:/Users/tusha/OneDrive/Desktop/DMWUI/Sigmoid/binary\_sentiment\_data.csv'

# Save the DataFrame to a CSV file

df.to\_csv(output\_file\_path, index=False)

def sigmoid(self, x):

return 1 / (1 + np.exp(-x))

**5.Support Vector Machine(SVM):**

def apply\_svm(self):

self.update()

self.update\_idletasks()

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Sigmoid/binary\_sentiment\_data.csv')

X\_numeric = df[['Text', 'Hashtags', 'Likes', 'Retweets', 'Month', 'Hour']]

y = df['Sentiment\_Binary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_numeric, y, test\_size=0.2, random\_state=42)

C\_values = [0.1, 1, 10, 100, 1000]

accuracy\_values = []

for C in C\_values:

svm\_classifier = SVC(kernel='rbf', C=C, gamma='scale')

svm\_classifier.fit(X\_train, y\_train)

y\_pred = svm\_classifier.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

accuracy\_values.append(accuracy)

sum = 0

flag = 0

for i in accuracy\_values:

sum = sum + i

flag = flag + 1

svm\_avg = sum/flag

print("SVM accuracy:",svm\_avg\*100)

self.graph\_ax.clear()

# self.graph\_ax.plot(C\_values, accuracy\_values, marker='o', linestyle='-')

# self.graph\_ax.set\_title('SVM Performance vs. Regularization Parameter (C)')

# self.graph\_ax.set\_xlabel('Regularization Parameter (C)')

# self.graph\_ax.set\_ylabel('Accuracy')

# self.canvas\_widget.draw()

plt.clf()

plt.plot(C\_values, accuracy\_values, marker='o', linestyle='-')

plt.title('SVM Performance vs. Regularization Parameter (C)')

plt.xlabel('Regularization Parameter (C)')

plt.ylabel('Accuracy')

plt.savefig('C:/Users/tusha/OneDrive/Desktop/DMWUI/svm\_performance.pdf')

messagebox.showinfo("Accuracy:", svm\_avg\*100)

**5.Random Forest:**

def apply\_random\_forest(self):

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Sigmoid/binary\_sentiment\_data.csv')

X\_numeric = df[['Text', 'Hashtags', 'Likes', 'Retweets', 'Month', 'Hour']]

y = df['Sentiment\_Binary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_numeric, y, test\_size=0.2, random\_state=42)

estimators\_values = range(1, 101)

accuracy\_values = []

for estimators in estimators\_values:

random\_forest = RandomForestClassifier(n\_estimators=estimators, random\_state=42)

random\_forest.fit(X\_train, y\_train)

y\_pred = random\_forest.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

accuracy\_values.append(accuracy)

sum = 0

flag = 0

for i in accuracy\_values:

sum = sum + i

flag = flag + 1

rf\_avg = sum/flag

print("Random Forest Accuracy:",rf\_avg\*100)

self.graph\_ax.clear()

plt.clf()

plt.plot(estimators\_values , accuracy\_values, marker='o', linestyle='-')

plt.title('Random Forest Performance vs. Number of Estimators ')

plt.xlabel('Number of Estimators')

plt.ylabel('Accuracy')

plt.savefig('C:/Users/tusha/OneDrive/Desktop/DMWUI/randomforest\_performance.pdf')

messagebox.showinfo("Accuracy:", rf\_avg\*100)

**5.Decision Tree:**

def apply\_decision\_tree(self):

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Sigmoid/binary\_sentiment\_data.csv')

X\_numeric = df[['Text', 'Hashtags', 'Likes', 'Retweets', 'Month', 'Hour']]

y = df['Sentiment\_Binary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_numeric, y, test\_size=0.2, random\_state=42)

max\_depth\_values = range(1, 21)

accuracy\_values = []

for depth in max\_depth\_values:

decision\_tree = DecisionTreeClassifier(max\_depth=depth, random\_state=42)

decision\_tree.fit(X\_train, y\_train)

y\_pred = decision\_tree.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

accuracy\_values.append(accuracy)

sum = 0

flag = 0

for i in accuracy\_values:

sum = sum + i

flag = flag + 1

dt\_avg = sum/flag

print("Decision Tree accuracy:",dt\_avg\*100)

self.graph\_ax.clear()

plt.clf()

plt.plot(max\_depth\_values, accuracy\_values, marker='o', linestyle='-')

plt.title('Decision Tree Performance vs. max\_depth ')

plt.xlabel('max\_depth')

plt.ylabel('Accuracy')

plt.savefig('C:/Users/tusha/OneDrive/Desktop/DMWUI/decisiontree\_performance.pdf')

messagebox.showinfo("Accuracy:", dt\_avg\*100)

**5.Naive Bayes:**

def NaiveBayes(self):

# Step 1: Read data

df = pd.read\_csv('C:/Users/tusha/OneDrive/Desktop/DMWUI/Sigmoid/binary\_sentiment\_data.csv')

# Step 2: Data Preparation

X\_numeric = df[['Text', 'Hashtags', 'Likes', 'Retweets','Month','Hour']] # Numerical features

y = df['Sentiment\_Binary'] # Replace 'Target\_Label' with the name of your target variable

# Step 3: Split data into training and testing sets

X\_numeric\_train, X\_numeric\_test, y\_train, y\_test = train\_test\_split(X\_numeric, y, test\_size=0.2, random\_state=42)

# Step 4: Initialize Naive Bayes classifier

naive\_bayes = GaussianNB()

# Step 5: Train the Naive Bayes model

naive\_bayes.fit(X\_numeric\_train, y\_train)

# Step 6: Model Evaluation

y\_pred = naive\_bayes.predict(X\_numeric\_test)

print(classification\_report(y\_test, y\_pred))

# Optionally, you can print the accuracy of the model

accuracy = naive\_bayes.score(X\_numeric\_test, y\_test)

print("Naive Bayes Accuracy:", accuracy\*100)

messagebox.showinfo("Accuracy:", accuracy\*100)

**5.GUI:**

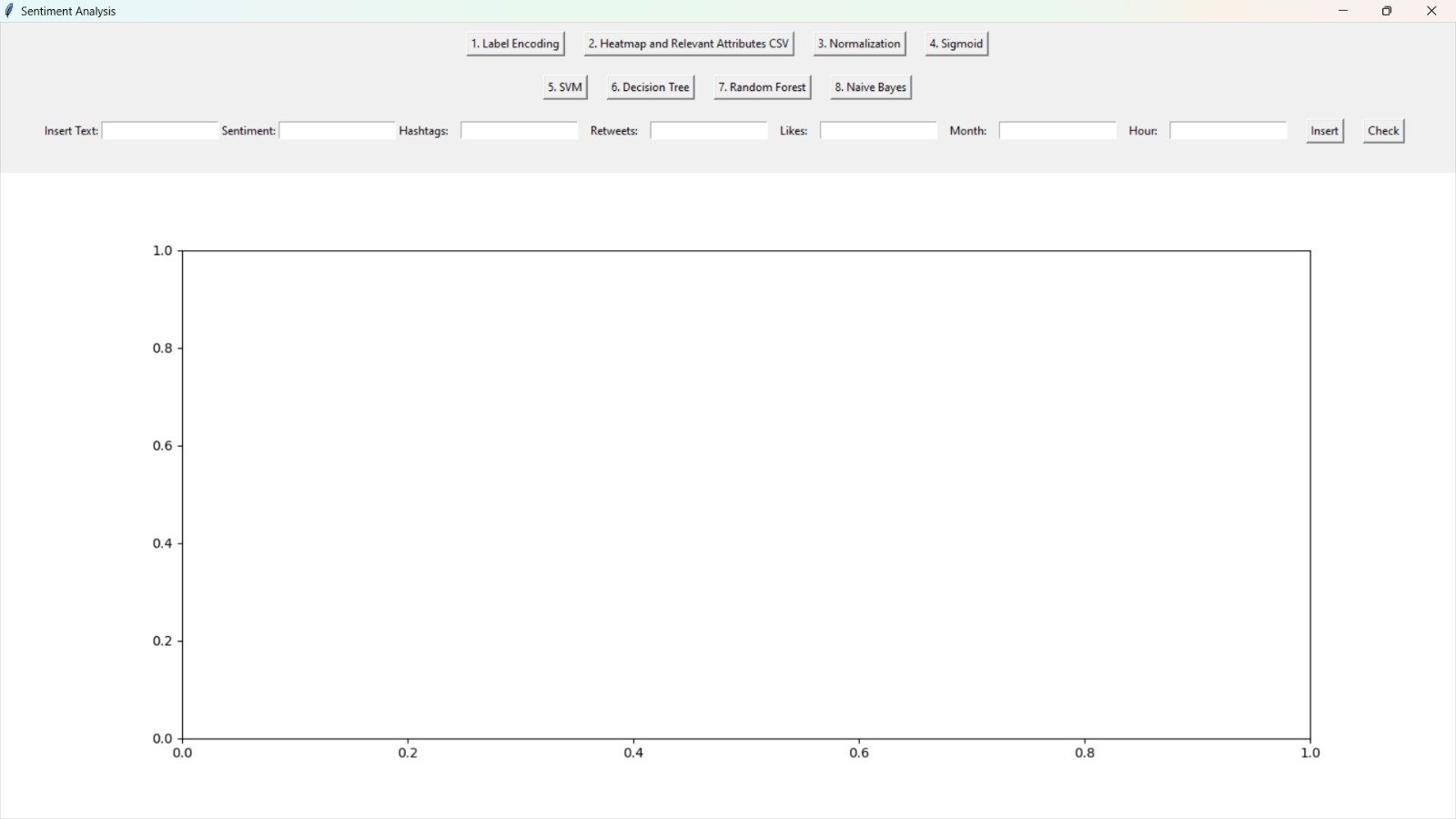
****

Fig. 7.3: Graphic User Interface

**Chapter 8**

**Result Analysis and Discussion**

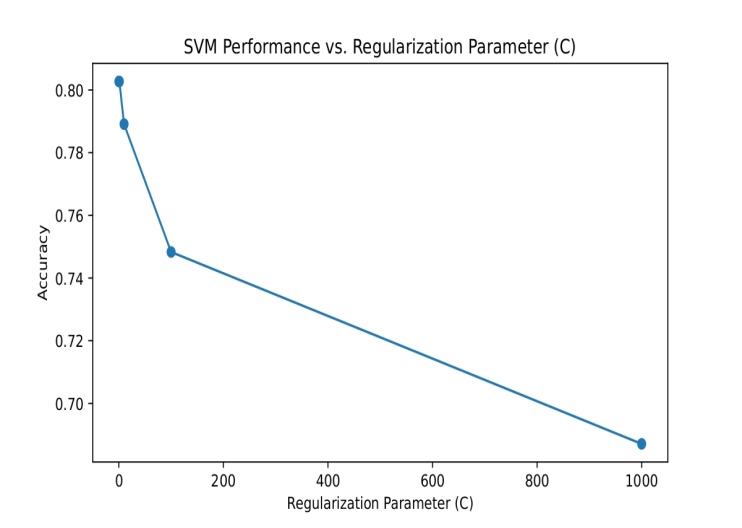
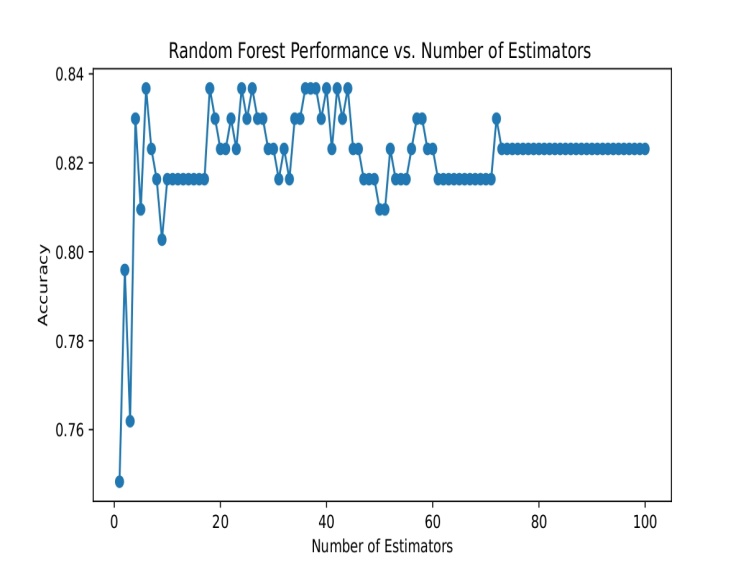
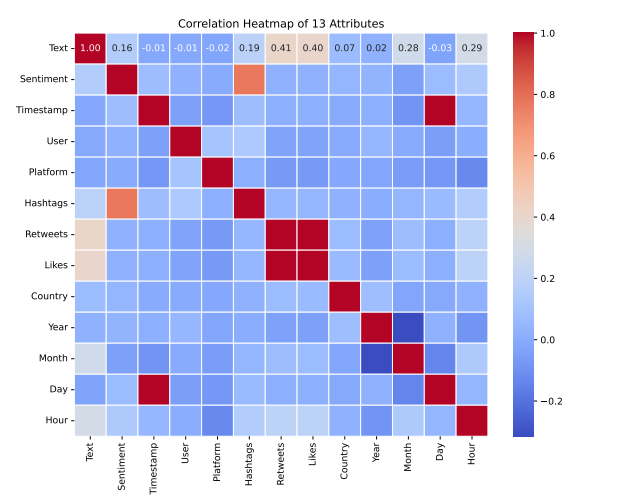


Fig. 8.2: Random Forest model performance

Fig. 8.1: SVM model performance



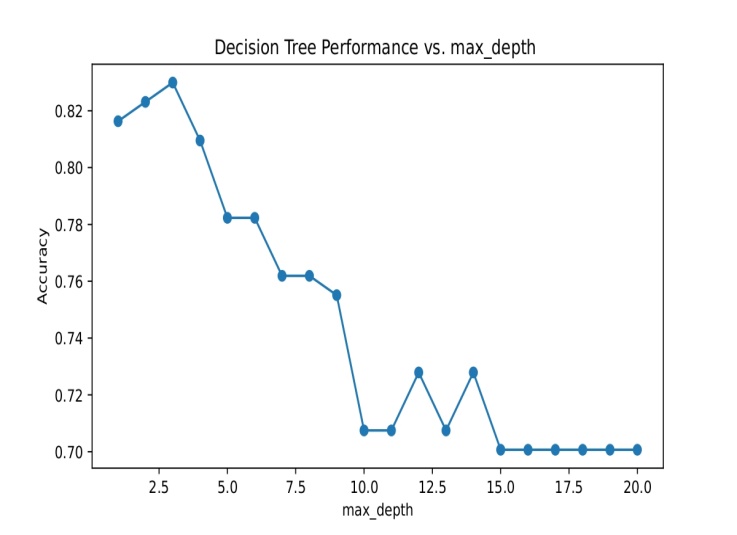


Fig.8.4. Correlation Heatmap

Fig.8.3. Decision Tree performance

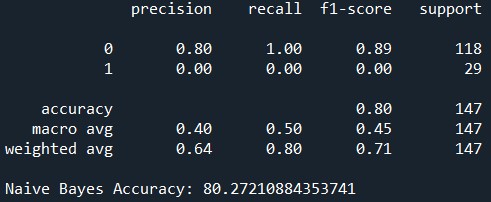


Fig. 8.5. Naïve Bayes Accuracy

**Conclusions**

In conclusion, social media sentiment analysis has become a powerful tool for businesses, governments, and researchers to gain valuable insights into public opinion and behavior. As the field continues to evolve, we can expect to see continued growth, advancements in AI and machine learning, and an increased focus on addressing the ethical considerations surrounding this technology. The future of sentiment analysis holds great promise, but also requires vigilance to ensure it is used responsibly and for the betterment of society.

**References**

[1] Research Paper Name: Sentimental Analysis of Twitter Comments on Covid-19

Author: Supriya Raheja and Anjani Asthana

[2]Research Paper Name: Sentimental Analysis using Capsule Network with Gravitational Search Algorithm

Author: V. Diviya Prabha and R. Rathipriya

[3]Research Paper Name: Performance of Sentimental Analysis by Studying and Mining Social Media using Parsing Technique

Author: Ms C. M. Sowmya, Ms. S. Deena and Dr. S. Anbuchelian.

[4] Research Paper Name: Analyzing Sentimental Influence of Posts on Social Networks

Author: Beiming Sun and Vincent TY Ng.

[5]Research Paper Name: A systematic review of social media-based sentiment analysis: Emerging trends and challenges

Author: Qianwen Ariel Xu , Victor Chang and Chrisina Jayne

[6] Research Paper Name: Sentiment Analysis for Social Media

Author: R. A. S. C. Jayasanka, M. D. T. Madhushani, E. R. Marcus, I. A. A. U. Aberathne and S. C. Premaratne.

[6] Research Paper Name: Sentiment Analysis on Social Media

Author: Federico Neri, Carlo Aliprandi, Federico Capeci and Montserrat Cuadros