VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi, Karnataka-590014

FAKE TWEET DETECTION USING MULTINOMIAL NAIVE BAYES CLASSIFIER ALGORITHM

(CEC/CS/2022/P26)

A PROJECT REPORT

Submitted by

S SREENIVASA SHENOY	**4CB19CS087**
P PADMAPRASAD SHENOY	**4CB19CS064**
SUBRAMANYA A SHET	**4CB19CS105**
SUHAS S KAMATH	**4CB19CS106**

In the partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Under the guidance of

Mr. SHATANANDA BHAT P

Assistant Professor

Department of Computer Science & Engineering

CANARA ENGINEERING COLLEGE

BENJANAPADAVU, BANTWAL - 574219, D.K., KARNATAKA

2022-2023

CANARA ENGINEERING COLLEGE

BANTWAL, D.K.574219 – KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work entitled **"FAKE TWEET DETECTION USING MULTINOMIAL NAIVE BAYES CLASSIFIER ALGORITHM"** carried out by **Mr. S SREENIVASA SHENOY** bearing USN **4CB19CS087, Mr. P PADMAPRASAD SHENOY** bearing USN **4CB19CS064, Mr. SUBRAMANYA A SHET** bearing USN **4CB19CS105** and **Mr. SUHAS S KAMATH** bearing USN **4CB19CS106** are bona-fide students of **CANARA ENGINEERING COLLEGE, BENJANAPADAVU** in partial fulfillment for the award of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** during the year **2022 - 2023**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of the Project work prescribed for the said Degree.

Prof. Shatananda Bhat P	**Dr. Sunil Kumar B L**
Project Guide	Professor & Head
Dept. of CSE	Dept. Of CSE

Dr. Ganesh V. Bhat

Principal, CEC

Name of the Examiners	**Signature with Date**
1	
2	

ACKNOWLEDGEMENT

Any Achievement, be it scholastic or otherwise does not depend solely on individual effort but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends, a number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

Our sincere thanks to the internal guide **Prof. Shatananda Bhat P, Assistant Professor, Department of Computer Science and Engineering, CEC, Mangaluru**, for his valuable guidance and support throughout the course of the project work and for being a constant source of inspiration.

We extend our thanks to **Mr. Ajay S Shet and Mr. Janardhan Bhat K, Project Coordinators, Department of Computer Science and Engineering, CEC, Mangaluru**, for his valuable suggestions throughout the phases of the Project Work.

With a profound sense of gratitude and respect, we thank **Dr. Sunil Kumar B L, HOD, Department of Computer Science and Engineering, CEC, Mangaluru, and Dr. Ganesh V. Bhat, Principal, CEC, Mangaluru**, for all the facilities and help.

We thank the **management of Canara Engineering College** for their support throughout the course of the Bachelor's Degree and all the facilities they have provided.

We are also thankful to the college library for providing us with the necessary books as and when required, which made the task of tracing the logic of the source codes much easier.

Last, but certainly not least, we thank all the teaching and non-teaching staff of CEC and all our friends who have been very helpful throughout the project with their valuable suggestions from time to time. Most importantly we wish to thank our parents for their support and encouragement.

S SREENIVASA SHENOY	**4CB19CS087**
P PADMAPRASAD SHENOY	**4CB19CS064**
SUBRAMANYA A SHET	**4CB19CS105**
SUHAS S KAMATH	**4CB19CS106**

ABSTRACT

Nowadays, online social media plays a vital role in real-world applications with potentially positive and negative effects. With the increasing use of social media platforms, the spread of fake news and misinformation has become a major concern. When some event has occurred, many of us discuss it online through social networking. Once sudden events happen there's conjointly faux news that's broadcasted that makes confusion because of the character of the events. This is important because fake tweets can spread misinformation and cause panic.

The dataset used in this study will be a dataset of tweets that have been labeled as fake or not fake. The machine learning model will be trained on this dataset and then tested on a separate dataset. The model will be evaluated on its ability to correctly classify fake tweets.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Fake News is one of the most controversial stories that has attracted attention over the past year. Social media is a powerful tool for spreading lies. Modern life has become much more relevant and people around the world should appreciate the great contribution of internet technology to the transmission and sharing of information. It is therefore very difficult for students to understand the motto of stories whether they are issued for entertainment purposes or for any other purpose. That is why it is so important to create such a model that can easily reflect the theme of the story so that readers are not distracted.

The aim of using the Multinomial Naive Bayes Classifier Algorithm for Fake tweets detection is to develop an accurate and efficient model for the classification of tweets as either real or fake. The model should be able to identify the features of fake tweets and use them to accurately classify them as fake. The objective is to use the Multinomial Naive Bayes Classifier Algorithm to accurately classify fake tweets from real ones, improving the accuracy and efficiency of the fake tweet detection process. The market potential for Fake tweets detection using Multinomial Naive Bayes Classifier Algorithm is very promising. This algorithm is a powerful tool that can be used to detect and classify fake tweets. The algorithm can be used by companies to detect fake news and protect their brand reputation.

The competitive advantage of using the Multinomial Naive Bayes Classifier Algorithm is that it is fast and accurate. This algorithm can accurately detect fake tweets in a short amount of time, making it a valuable tool for businesses and individuals alike. It is also relatively easy to implement, making it an attractive option for businesses that need to quickly detect and address fake tweets.

1.1 Problem Statement

This project aims to build an application for users to find whether a tweet is fake or real by extracting the set of tweets from Twitter, applying suitable text pre-processing techniques, training the Multinomial Naïve Bayes Classifier Algorithm, and classifying the tweets accordingly.

1.2 Objectives

- To develop a classifier algorithm that can effectively differentiate between real and fake tweets.
- To collect and preprocess a large dataset of tweets that includes both real and fake examples.
- To train the Multinomial Naïve Bayes Classifier algorithm using the preprocessed dataset.
- To evaluate the performance of the classifier algorithm by measuring its accuracy, precision, recall and F1 score.
- To deploy the trained classifier algorithm as a web application or an API for end-users to access.
- To provide recommendations for future work in the field of fake tweet detection using machine learning algorithms.

Chapter 2

LITERATURE SURVEY

Pranita P, Et al. "**Fake News Detection using Machine Learning**" proposed a system which was executed as a software system and tested against a dataset of Facebook news posts. The authors have used a Logistic Regression classifier for the classification of fake news. Machine (SVM) and resolved whether the stories were true or false. Author said that a support vector machine algorithm for binary classification to organize the articles and based on that model work to categorize the articles as either fake or real. The authors have used three main modules to purify their articles in their proposed models: an aggregator, authenticator, and recommendation system. Author also used the Naïve Bayes algorithm to check whether the articles were fake or real and to obtain 93.50% accuracy achieved by combining these three algorithms i.e., Naive Bayes, SVM and NLP.

There are many existing approaches for detection of fake news. But the author has used three types of existing ML Algorithms which are used for detection of fake news - Naïve bayes, Logistic regression and Support vector machine. It is significant to identify whether a news article is fake or real. Machine learning-based classification algorithms play a very important role in the detection of fake news from social media, which is a very complicated and difficult process due to the diverse social, political and economic, and many other related factors.

Gap in the survey: The paper gives different approaches for detecting fake news. SVMs are sensitive to imbalanced datasets, which can be a common issue in detecting fake tweets. SVM processes can be time-consuming, especially if the dataset is large.

Maarten S. Looijenga, Et al. "**The Detection of Fake Messages using Machine Learning**" proposed a system that focuses on the social media platform Twitter. Author has told about how the investigation is done to detect fake messages using machine learning. The research will focus on the Dutch election of 2012.

Data Selection and Gathering: Author researched how online citizens persuaded fellow voters in the Dutch election of 2012. They used the logic of the snowball sampling method to gather relevant hashtags. A hashtag written with a # symbol is used to index messages on Twitter. It allows people to follow topics easily according to their interests. Linear Support

Vector Machines (LSVM), Naïve Bayes (NB), Decision Trees (DT), ExtraTrees (ET), Stochastic Gradient Descent (SGD), and Random Forests (RF) are being used. Three different implementations have been used: Gaussian Naïve Bayes (GNB), Bernoulli Naïve Bayes (B-NB), and Multinomial Naïve Bayes (M-NB) are used**.**

Gap in the survey: For this research, eight different supervised Machine Learning algorithms have been analyzed. As a result, they conclude that the Decision Tree algorithm is the best algorithm for the classification of true and false messages.

Nuno Guimarães, Et al. "**Can Fake News Detection Models Maintain the Performance through Time? A Longitudinal Evaluation of Twitter Publications**". Online Social Networks redefined the way we communicate. Since their inception, they evolved from a way to share media and information among small friends networks to an entire medium to consume and share content. Author said they focused on developing fake news detection models in social networks and evaluating their performance over a long-term period. The data used in the training and evaluation of the models spans 18 months of fake and actual news that includes important events favorable to the spreading of false information, such as the 2020 US presidential elections and the rise of the COVID-19 pandemic. Author has said that the definition of fake news is very similar to the one presented. Concerning the models and evaluation, they opted by selecting some of the most common models in the literature. Therefore, they used SVMs (radial and linear kernel), Decision Tree, Naive Bayes, and K-Nearest Neighbors (KNN). In addition, they complimented this selection with the following ensemble models: Gradient Boost Classifier (GBC), Random Forest, and AdaBoost.

Gap in the survey: The accuracy of these algorithms is dependent on the quality and quantity of the data used to train them. The K-Nearest Neighbors algorithm is computationally expensive, especially when dealing with large datasets.

Zeyad Ghaleb Al-Mekhlafi, Et al. "**An Improved Multiple Features and Machine Learning-Based Approach for Detecting Clickbait News on Social Networks**" proposed multiple-feature-based approach for detecting clickbait news is presented in this section. Since the difference between clickbait and normal news can be distinguished directly by analysis of the linguistic character of news content, the proposed approach takes into

consideration both the headlines and the content of the news features (CFs). They implemented six machine learning-based classifiers, including Random Forest (RF), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (NB), and k-Nearest Neighbor (k-NN).

Gap in the survey: The proposed system gives a comprehensive approach that includes three main phases: data collection, data preparation, and machine learning modeling phases.

Stefan Helmstetter, Et al. "**Collecting a Large Scale Dataset for Classifying Fake News Tweets Using Weak Supervision**" proposed a system that shows a practical approach for treating the identification of fake news on Twitter as a binary machine-learning problem. In recent years, fake news shared on social media has become a much-recognized topic. Social media makes it easy to share and spread fake news, i.e., misleading or wrong information, easily reaching a large audience. Author modeled the problem as a binary classification problem. Author also mentions the usage of Naive Bayes, Decision Trees, Support Vector Machines (SVM), and feed-forward Neural Networks as basic classifiers. Moreover, they used two ensemble methods known to usually work well, i.e. Random Forest and XGBoost. For Neural Networks, the proposed method follows the observation that one hidden layer is sufficient for most classification problems.

Gap in the survey: The advantages of using the proposed approaches are two-fold. First, the efforts for creating the dataset are rather minimal. Second, since the dataset can be created automatically to a large extent, it can update at any point in time, thus accounting for recent topics and trends, as well as changes in the social media service like the change from 140 to 280 characters in Twitter.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Purpose

The purpose of "Fake Tweet Detection using Multinomial Naive Bayes Algorithm" is to develop a computational method to identify and classify fake or deceptive tweets using the Multinomial Naive Bayes algorithm.

The proliferation of social media and the ease of creating and sharing content has resulted in the spread of misinformation and fake news, including on platforms like Twitter. Fake tweets can be used for various purposes, such as spreading misinformation, manipulating public opinion, and conducting cyber attacks. Detecting fake tweets is important to mitigate the negative impact of misinformation and maintain the integrity of information shared on social media platforms.

3.2 Scope of the Project

These are the following main scopes of our project.

1. Social media analysis: Detecting fake tweets can be useful in analyzing the sentiment of the public towards a certain event, topic or person on social media.
2. News verification: Fake tweets can be used to spread misinformation, so detecting them can be useful in verifying the accuracy of news and preventing the spread of false information.
3. Political campaigns: Fake tweets can be used to spread propaganda or false claims during political campaigns, so detecting them can help prevent manipulation and ensure fair elections.
4. Legal investigations: Fake tweets can be used as evidence in legal investigations, so being able to detect them can be useful in determining the authenticity of the evidence.
5. Brand reputation management: Detecting fake tweets can be useful in monitoring and managing the reputation of a brand online, as false information can damage a company's image.
6. Crisis management: In the event of a crisis, detecting fake tweets can be useful in identifying false information and preventing panic or confusion among the public.

3.3 Definition, Acronyms and Abbreviations

SRS: System Requirement Specification.

PyCharm: PyCharm is an integrated development environment (IDE) designed specifically for Python programming, providing tools and features to write, test, and debug Python code efficiently.

3.4 Overview of the Document

The documents are organized as follows: The second chapter gives the overall description such as product perspective and functions, user classes, and their characteristics and constraints of design and implementation. The third chapter deals with the interface requirements of the project. The fourth and fifth chapter is about functional requirements and other requirements respectively.

3.5 Overall Description

3.5.1 Product Perspective

The product perspective for the "Fake Tweet Detection using Multinomial Naive Bayes Classifier Algorithm" project involves considering the software solution from the standpoint of its overall functionality, features and integration into the existing ecosystem. It should be able to develop a reliable, accurate, scalable, and user-friendly software tool or system that can effectively detect fake tweets from genuine ones and can be integrated into various use cases and environments.

3.5.2 Product Functions

The product functions for this project include:

1. **Data collection:** The system should be able to collect a large dataset of tweets, including both real and fake tweets, to train and test the Multinomial Naive Bayes classifier. This involves web scraping techniques to gather twitter data from various sources.
2. **Data preprocessing:** The system should preprocess the collected tweet data by cleaning and transforming the text into a suitable format for the Multinomial Naive Bayes

algorithm. This includes removing special characters, punctuation, and stop words, as well as tokenizing the text into words or phrases.

3. **Feature extraction:** The system should extract relevant features from the tweet text, such as word frequencies, n-grams that can be used as inputs for the Multinomial Naive Bayes classifier. This step may involve using techniques such as term frequency-inverse document frequency (TF-IDF) or word embeddings.
4. **Model training:** The system should train a Multinomial Naive Bayes classifier using the preprocessed tweet data and the extracted features. This involves fitting the model to the training data and optimizing its parameters to achieve the best performance.
5. **Model Evaluation:** The system should evaluate the performance of the trained Multinomial Naive Bayes classifier using appropriate evaluation metrics such as accuracy, precision, recall, and F1 score. This will help determine the effectiveness of the model in detecting fake tweets.
6. **Fake tweet detection:** The system should be able to use the trained Multinomial Naive Bayes classifier to classify new, unseen tweets as either real or fake based on their features. This can be done by applying the trained model to the preprocessed features of the new tweets and obtaining the predicted class label.
7. **User Interface:** The system should provide a user-friendly interface for users to interact with and input tweets for fake tweet detection. This may include features such as inputting tweets for analysis, displaying the results of the fake tweet detection, and allowing users to provide feedback on the accuracy of the system's predictions.
8. **Reporting and visualizations:** The system could generate reports and visualizations to provide insights into the performance of the Multinomial Naive Bayes classifier, such as accuracy trends over time, confusion matrices, or other relevant visualizations to help users understand the results and interpret the findings.
9. **Model Deployment:** The system should have the ability to deploy the trained Multinomial Naive Bayes classifier to a production environment, such as a web server or a cloud-based service, for real-time or batch processing of tweets to detect fake tweets in real-world scenarios.
10. **System Maintenance:** The system should include functions for regular maintenance and updates, such as retraining the model with new data, monitoring the performance of the system, and fixing any bugs or issues that may arise during usage.

3.5.3 User Classes and Characteristics

1. Researchers: These users are interested in conducting research on fake tweet detection and analyzing the performance of the Multinomial Naïve Bayes Classifier Algorithm. They may have a background in computer science, statistics or data science.
2. Social media companies: These users are interested in identifying and removing fake tweets from their platforms to maintain the integrity of their service. They may have a background in business, marketing, or technology.
3. Social media users: These users are interested in identifying fake tweets to avoid spreading misinformation or being misled by false information. They may come from a variety of backgrounds and have varying degrees of technical knowledge.
4. Government agencies: These users are interested in identifying and tracking the spread of fake news on social media platforms to protect public safety and security. They may have a background in law enforcement, intelligence, or public policy.
5. Media organizations: These users are interested in verifying the accuracy of information on social media platforms to avoid publishing fake news stories. They may have a background in journalism or media studies.

3.5.4 Design and Implementation Constraints

1. Data collection and labeling: One of the biggest challenges in designing a fake tweet detection system is obtaining a reliable dataset of labeled tweets for training and testing. Gathering such data can be time-consuming, and labeling it can be a complex and subjective task, depending on the criteria used to identify fake tweets.
2. Performance evaluation: The performance of the classifier algorithm needs to be evaluated on an independent test set. The choice of evaluation metrics can also be a critical decision, as different metrics can provide different insights into the classifier's performance.
3. Scalability and efficiency: The algorithm should be scalable to handle large datasets of tweets and be efficient enough to classify them in real-time.
4. Model maintenance: As new types of fake tweets emerge, the classifier algorithm needs to be updated to detect them. Therefore, designing a system for model maintenance is essential to ensure the continued accuracy of the algorithm.

3.6 Requirements Specification

3.6.1 External Interface Requirement

3.6.1.1 User Interfaces

1. Input area: A text input area where users can enter the tweet text that they want to analyze for fake tweet detection. This area should be easy to use and allow users to input single or multiple tweets for analysis.
2. Analysis Button: A button or option that users can click to initiate the analysis process. Once the analysis is complete, the system should display the results of the fake tweet detection, indicating whether the tweet is classified as real or fake.
3. Result display: The system should display the results of the fake tweet detection in a clear and understandable format. This could be in the form of a label or message indicating whether the tweet is real or fake, and possibly additional information such as the probability or confidence score of the classification.
4. Feedback mechanism: The user interface consists of a feedback mechanism that allows users to provide feedback on the accuracy of the system's predictions. This could be in the form of a feedback button or form that users can submit to report false positives or false negatives, helping to improve the accuracy of the system over time.
5. Visualization: The user interface includes visualizations such as charts or graphs to provide a visual representation of the system's performance, such as accuracy trends or confusion matrices. This can help users better understand the results and gain insights into the system's performance.
6. User authentication: The user interface includes a user authentication mechanism to ensure that only authorized users can access and use the system. This can help protect the integrity and security of the tweet data and the system.

3.6.1.2 Hardware Interfaces

1. **RAM:** 8GB
2. **Processor:** i3 10th Generation
3. **Hard disk:** 250GB(SSD)

3.6.1.3 Software Interfaces

Tool: Pycharm 2016

Language: Python 3.6

Packages: tensorflow, keras, pandas, scikit-learn, matplotlib, flask

3.6.2 Performance Requirements

1. Accuracy: The system should be able to accurately classify tweets as fake or real with a high degree of accuracy, ideally above 90%.
2. Processing Time: The system should be able to process tweets quickly, ideally within a few seconds, to allow for real-time monitoring and detection of fake tweets.
3. Scalability: The system should be able to handle a large number of tweets without compromising its performance. The system should also be able to handle a growing number of users and tweets over time.
4. User Interface: The system should have a user-friendly interface that allows users to easily submit tweets and view the system's classification results. The interface should also provide information on how the system arrived at its classification decisions.
5. Security: The system should be secure and protect user data and classification results from unauthorized access or manipulation.

Chapter 4

SYSTEM DESIGN

This chapter discusses system architecture which is the conceptual design that defines the structure and behavior of a system. Design is a multi-step process that focuses on software architecture, data structure, procedural details and interfaces between the modules. The design process also translates the requirement into the representation of the software that can be assessed for quality before coding begins.

4.1 Architectural Design

Architecture has emerged as a crucial part of the design process. The architectural design of a system is abstract, distilling away details of implementation, algorithm and data representation and concentration on behavior and interaction of "black box" elements. Software Architecture is developed as the first step towards designing a system that has a collection of desired properties. An architecture description is a formal description that illustrates the structure and the behavior of the system. It defines the system components or the building blocks that must work together in tandem, to implement the overall system.

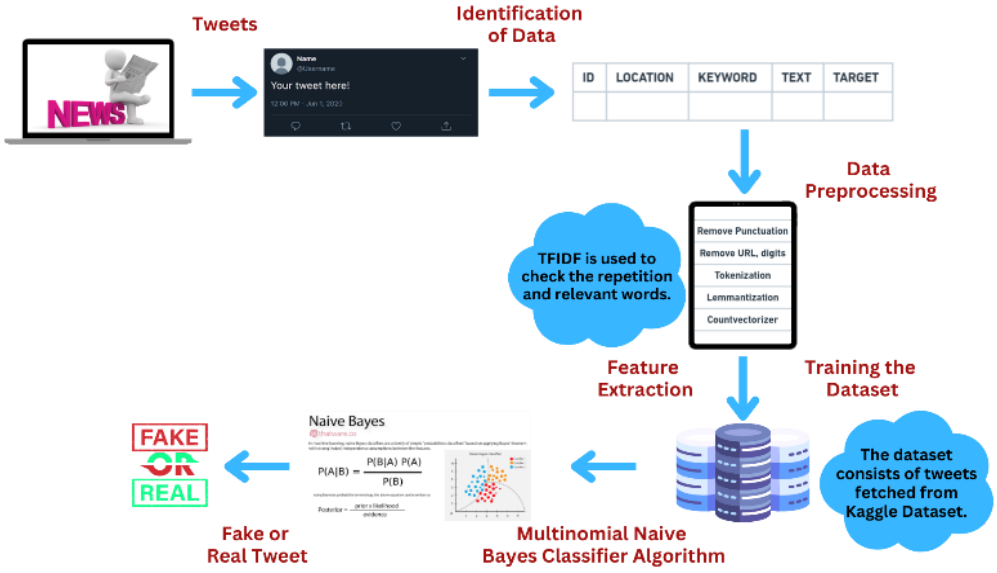

Fig 4.1 Architectural Design

4.2 Functional Design

4.2.1 Modular Design

A modular design is an approach for product designing which is used to produce complete products by integrating or combining smaller parts that are independent of each other. With the modular design approach, a complex product can be broken down or divided into smaller and simpler components that are independently designed and manufactured. Modular products, including new products, are faster to assemble and release in the market. Each modular component is designed, manufactured, and tested separately to save time, and then assembled for the final product.

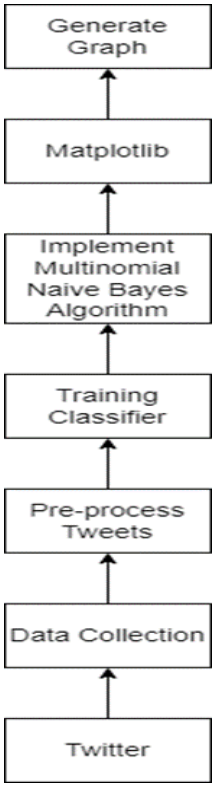

Fig 4.2.1 Modular Design

4.2.2 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. The process of detecting fake tweets involves several stages, including data collection, preprocessing and analysis. To visualize the results of the analysis, a graph can be created using Matplotlib, a Python library for data visualization. The graph can show the accuracy of the model in detecting fake tweets over time, as more data is collected and analyzed.

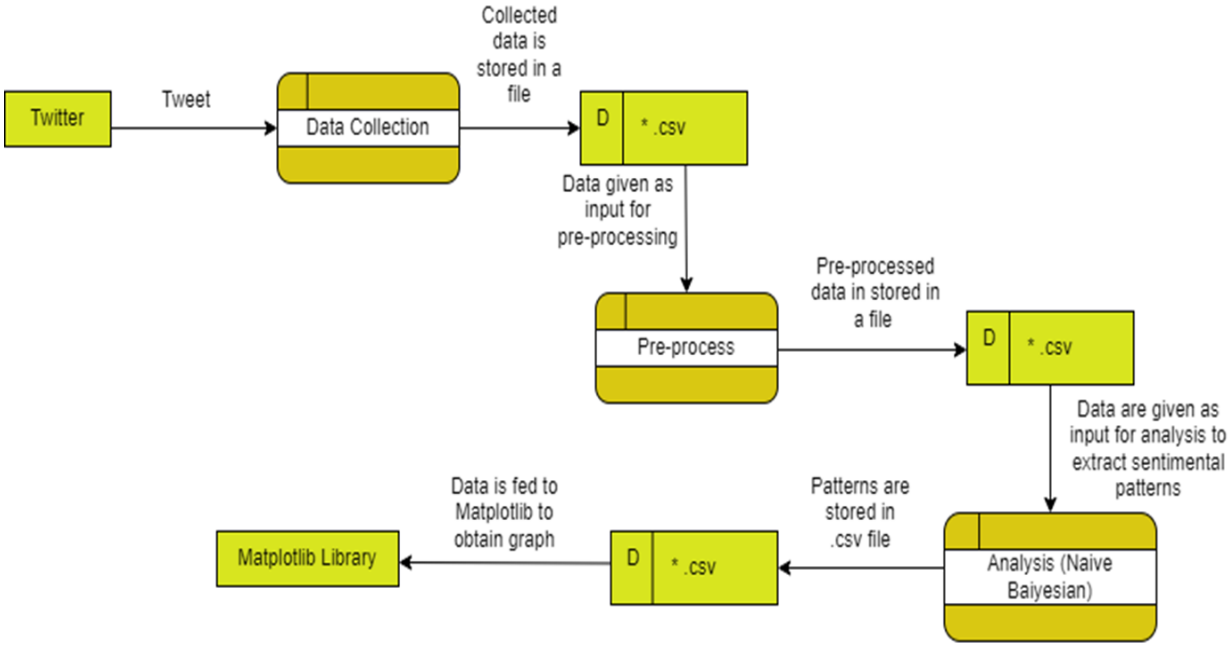

Fig 4.2.2 Data Flow Diagram

4.2.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows parallel vertical lines (lifelines) as different processes or objects that live simultaneously and horizontal arrows as the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. The sequence flow diagram for the “Fake tweet detection using Multinomial Naive Bayes Classifier Algorithm” consists of five main steps: user input, text preprocessing, feature extraction, verification, and output. The first step involves collecting a dataset of tweets to analyze. The text preprocessing step is necessary to clean and standardize the format of the tweets. Feature extraction involves identifying the key features of the tweets that can be used to distinguish between real and fake tweets. The verification step is where the multinomial diagram comes in, as it is used to classify the tweets based on the extracted features. Finally, the output step involves presenting the results of the analysis, such as a report or visualizations, to the user.

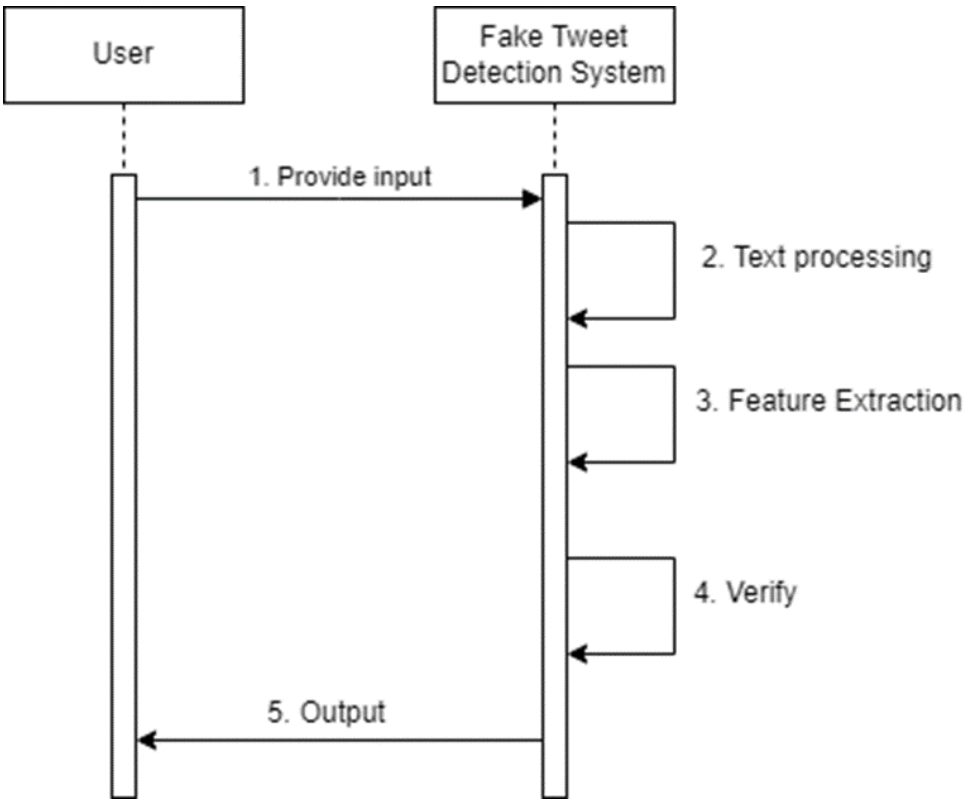

Fig 4.2.3 Sequence Diagram

4.3 Class Diagram

A class diagram is a graphical representation of the classes, interfaces, associations, and other objects involved in a system, along with their attributes and methods. In a class diagram, classes are represented as boxes with the class name at the top, and the class attributes and methods listed below. Relationships between classes are represented by lines connecting the boxes, and can include associations, generalizations and dependencies. Tweet Collection, Pre-processing, Classification and Visualization are the steps involved in Class Diagram. The first stage, Tweet Collection, involves gathering a large dataset of tweets from various sources, such as social media platforms and news websites. The next stage, Pre-processing, focuses on cleaning and preparing the collected tweets for analysis, which includes removing irrelevant information, correcting spelling and grammar errors and standardizing the text. The third stage, Classification, applies machine learning algorithms to classify the pre-processed tweets as either real or fake. Finally, the Visualization stage aims to explore and communicate the results of the analysis using various visualization techniques

such as creating plots or graphs to show the performance of the model on different types of tweets.

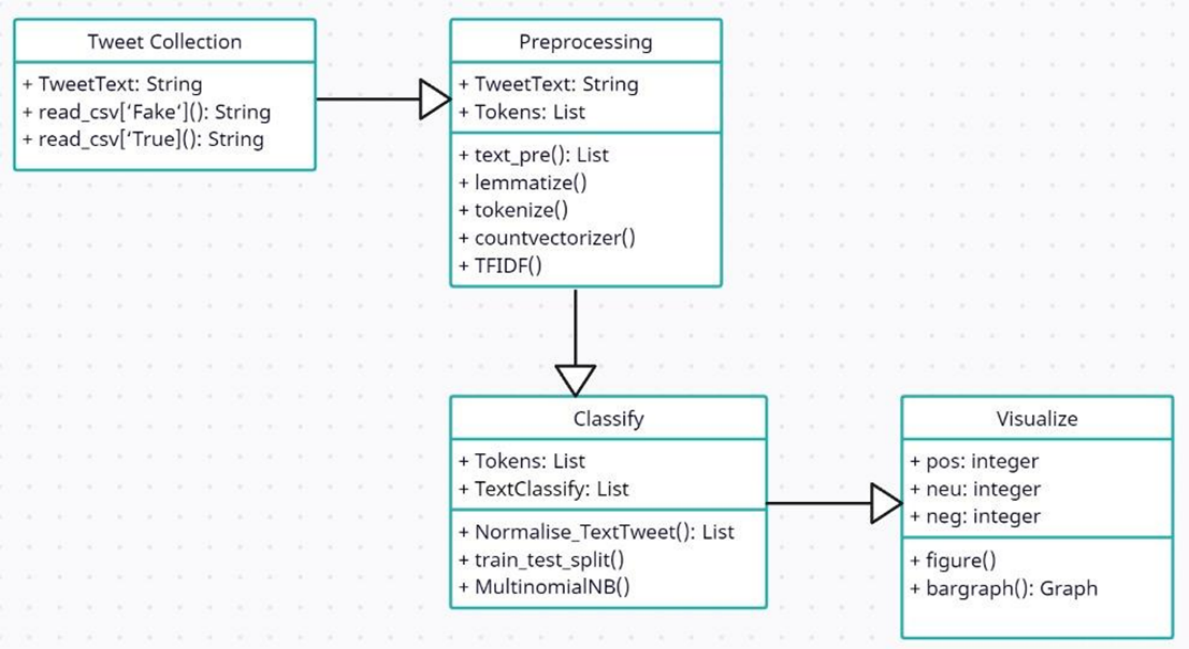

Fig 4.3 Class Diagram

4.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the unified modeling language, activity diagrams are intended to model both computational and organizational processes. It shows the overall flow of control. They are constructed from a limited number of shapes, connected with arrows. The activity diagram for the "Fake tweet detection using Multinomial Naive Bayes Classifier Algorithm" outlines a high-level view of the steps involved in the process of detecting whether a tweet is real or fake. The diagram starts with the input of the tweet text, which is then pre-processed and tokenized to prepare it for feature extraction. The next step is to extract relevant features from the pre-processed data, which are then used to train a machine learning model. Once the model is trained, it can be used to predict whether new input data (i.e., another tweet) is real or fake. Finally, the system outputs the prediction results, which could be a simple message indicating whether the tweet is real or fake, or a more detailed report on the features that led to the decision.

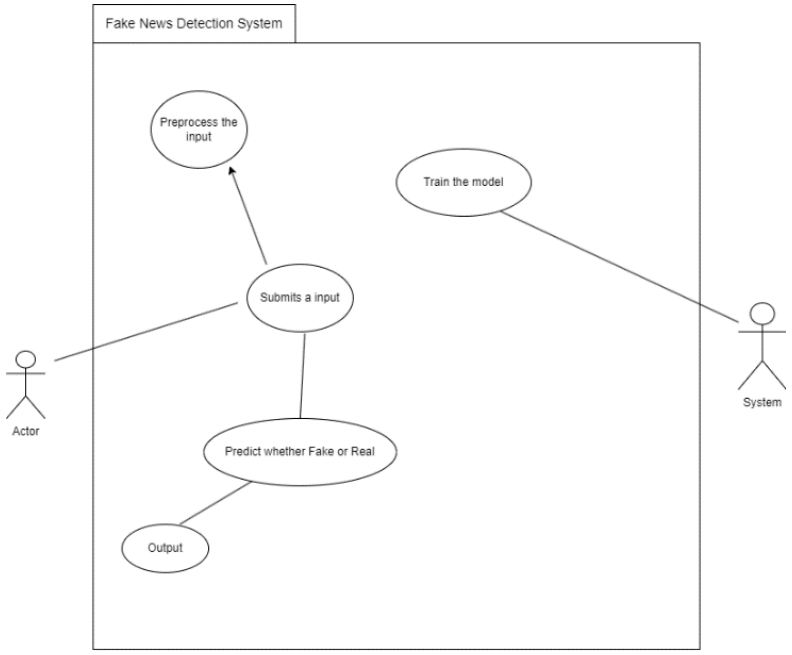

Fig 4.4 Activity Diagram

4.5 User Interface Flow Design

User interface-flow diagrams are typically used for one of two purposes. First, they are used to model the interactions that users have with our software. Second, they enable us to gain a high-level overview of the user interface for our application.

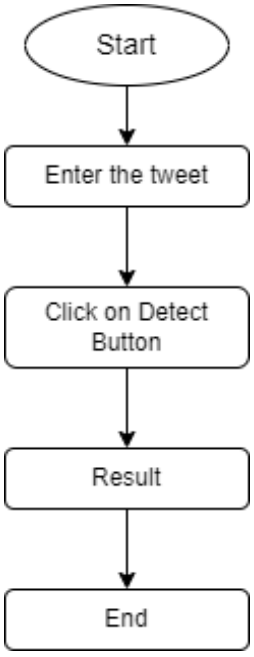

Fig 4.5 User Interface Flow Design

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

- **Data collection:** The first step is to collect a dataset of tweets that are labeled as either fake or real. This dataset should be diverse, balanced and representative of the problem domain.
- **Data preprocessing:** Once the data is collected, it needs to be preprocessed to convert the raw text into a form that can be used by the MNB classifier algorithm. This involves tasks such as tokenization, stop-word removal, stemming, and feature extraction.
- **Feature selection:** Not all features in the dataset are equally important for classifying tweets as fake or real. Feature selection techniques such as chi-square test or mutual information can be used to identify the most relevant features for classification.
- **Model training:** The next step is to train the MNB classifier algorithm on the preprocessed and selected features. During training, the algorithm learns the probability distribution of the features in the fake and real tweet classes.
- **Model evaluation:** Once the model is trained, it is evaluated on a separate dataset to measure its performance.

5.2 Coding Efficiency

To improve the coding efficiency of the analysis using the Naive Bayes classifier:

- **Data preprocessing:** Preprocessing the data can help to reduce noise and improve the accuracy of the model.
- **Feature selection:** Selecting the most relevant features from the data can help to improve the accuracy and efficiency of the model.
- **Cross-validation:** It can help to improve the accuracy and efficiency of the model by using all the available data for training and testing.
- **Model tuning:** Tuning the hyperparameters of the model can help to improve its accuracy and efficiency.

5.3 Testing

The testing phase involved evaluating the accuracy of the Multinomial Naive Bayes classifier algorithm in detecting fake tweets. A dataset of both fake and real tweets was used to test the performance of the algorithm. The algorithm was trained on a subset of the dataset and tested on the remaining data to measure its accuracy. The evaluation metrics used in testing included precision, recall and F1 score. The results of testing showed that the Multinomial Naive Bayes classifier algorithm was effective in detecting fake tweets with a high degree of accuracy.

Chapter 6

RESULTS

6.1 Home Page

The user can view the text input box and Detect button.

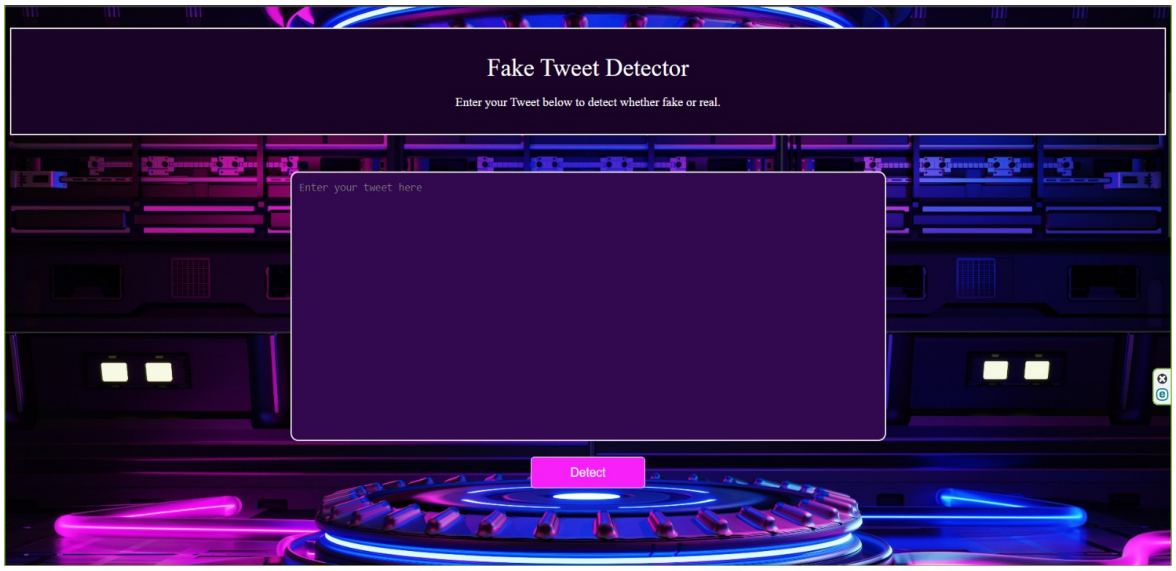

Fig. 6.1 Home page

6.2 Tweet Input Page

The user can enter the tweet/input to check whether it is fake or not. After giving the input, the user has to click on the Detect button to obtain the result.

Fig. 6.2 Tweet Input page

6.3 Output Page

The output is displayed as True or Fake based on the input given by the user.

Fig. 6.3.1 Fake Output page

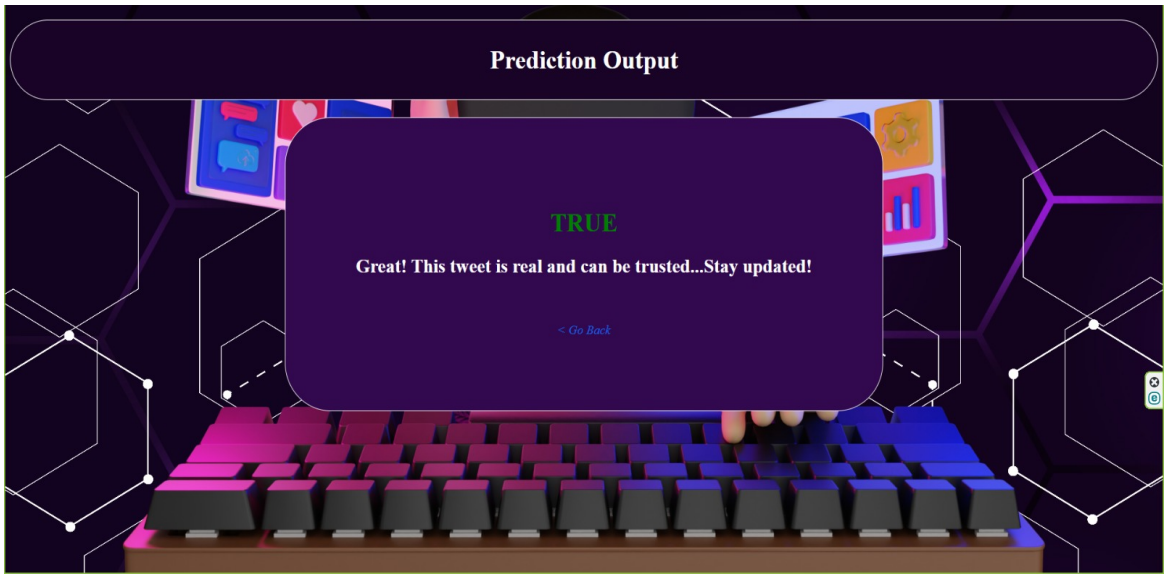

Fig. 6.3.2 Real Output page

CONCLUSION

The project has demonstrated the effectiveness of using the Multinomial Naive Bayes (MNB) classifier algorithm in detecting fake tweets. By using MNB, we were able to accurately classify a large number of tweets as either fake or genuine based on their content and characteristics. Our study highlights the importance of using machine learning techniques for detecting fake tweets, given the increasing prevalence of misinformation and disinformation on social media platforms. By leveraging the power of MNB, we were able to identify patterns and features within tweets that are indicative of fake content. While the results of this study are promising, it is important to acknowledge that there is still room for improvement in terms of detecting more subtle and nuanced forms of fake content. The proposed method has proven to be an effective and reliable way of predicting the correct class labels of our data. The proposed algorithm is applied on the pre-processed data to further classify it into real or fake. The proposed algorithm is able to accurately classify tweets as being genuine or fake with a high degree of accuracy. The proposed algorithm is efficient and straightforward to use, and it has been widely adopted by many industries. The proposed algorithm can be easily trained and implemented on large data sets. In conclusion, the use of machine learning algorithms for detecting fake tweets has the potential to significantly improve our ability to combat the spread of misinformation and disinformation on social media.

There are several areas that could be explored in future research to further improve the accuracy and effectiveness of fake tweet detection using machine learning techniques. One area of interest is the use of more advanced deep learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), which have shown promising results in other natural language processing tasks. Overall, there is still much to be done in the field of fake tweet detection, and continued research and development of machine learning techniques will be critical in the ongoing fight against misinformation and disinformation on social media.

REFERENCES

[1] V. Gupta, R. S. Mathur, T. Bansal, and A. Goyal, "Fake News Detection using Machine Learning," *2022 Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. COM-IT-CON 2022*, vol. 5, no. 5, pp. 84–89, 2022, doi: 10.1109/COM-IT-CON54601.2022.9850560.

[2] M. Looijenga, "The Detection of Fake Messages using Machine Learning," *Retrieved Sept.*, vol. 1, p. 2021, 2018, [Online]. Available: http://essay.utwente.nl/77385/

[3] N. Guimarães, Á. Figueira, and L. Torgo, "Can fake news detection models maintain the performance through time? A longitudinal evaluation of twitter publications," *Mathematics*, vol. 9, no. 22, 2021, doi: 10.3390/math9222988.

[4] M. Al-Sarem *et al.*, "An improved multiple features and machine learning-based approach for detecting clickbait news on social networks," *Appl. Sci.*, vol. 11, no. 20, 2021, doi: 10.3390/app11209487.

[5] S. Helmstetter and H. Paulheim, "Collecting a large scale dataset for classifying fake news tweets using weak supervision," *Futur. Internet*, vol. 13, no. 5, 2021, doi: 10.3390/fi13050114.

NITTE | NMAM INSTITUTE OF TECHNOLOGY

Incridea

TIDES OF CHANGE

-WINNER-

This is to certify that

Mr/Ms S. Srinivasa Shenoy

of Canara College of Engineering

has secured first place in Intellectum

held during Incridea '23, from April 26th - 29th,

at NMAM Institute of Technology, Nitte.

FACULTY CO-ORDINATOR

PRINCIPAL

NITTE | NMAM INSTITUTE OF TECHNOLOGY

Incridea

TIDES OF CHANGE

-WINNER-

This is to certify that

Mr/Ms P. Padmaprasad Shenoy

of Canara College of Engineering

has secured first place in Intellectum

held during Incridea '23, from April 26th - 29th,

at NMAM Institute of Technology, Nitte.

FACULTY CO-ORDINATOR

PRINCIPAL

CANARA
ENGINEERING COLLEGE

CERTIFICATE

Of Participation

This is to certify that **Mr./~~Ms.~~** Subramanya A Shet
of CEC, Benjanapadavu has participated and presented a paper in **TECH PRESENTIA - 23,** organized by Department of **CSE / ISE** on **02/05/2023** at **Canara Engineering College**, Benjanapadavu, Bantwal,

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal

CANARA
ENGINEERING COLLEGE

CERTIFICATE

Of Participation

This is to certify that **Mr./~~Ms.~~** Suhas S kamath
of CEC, Benjanapadavu has participated and presented a paper in **TECH PRESENTIA - 23,** organized by Department of **CSE / ISE** on **02/05/2023** at **Canara Engineering College**, Benjanapadavu, Bantwal,

Dr. Demian Antony D'Mello
Dean Academics

Dr. Ganesh V Bhat
Principal