VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590014

FAKE TWEET DETECTION USING MULTINOMIAL NAIVE BAYES CLASSIFIER ALGORITHM (CEC/CS/2022/P26)

SOFTWARE REQUIREMENTS SPECIFICATION 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

This is to certify that "**Mr. S SREENIVASA SHENOY(4CB19CS087), Mr. P PADMAPRASAD SHENOY (4CB19CS064), Mr. SUBRAMANYA A SHET (4CB19CS105) and Mr. SUHAS S KAMATH (4CB19CS106)"** has successfully completed the PHASE 2 of the project entitled **"FAKE TWEET DETECTION USING MULTINOMIAL NAIVE BAYES CLASSIFIER ALGORITHM"** under the guidance of **Prof. SHATANANDA BHAT P**. The project report has been approved as it satisfies the academic requirements in respect of Project work.

Signature of the Guide

(**Prof. Shatananda Bhat P)**

TABLE OF CONTENTS

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 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 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.

1.2 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.

1.3 Definitions, Acronyms & 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.

1.4 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.

Chapter 2

OVERALL DESCRIPTION

2.1 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. Some key aspects of the product perspective for this project include:

1. **Accuracy**: The primary goal of the project would be to develop a highly accurate fake tweet detection system. The product should strive for high precision and recall rates to minimize false positives and false negatives in the classification results.
2. **Scalability:** The product should be designed to handle a large volume of tweets in real-time or batch processing scenarios. It should be able to efficiently process and analyze tweets from various sources, such as social media streams, news feeds, or custom datasets, to detect fake tweets in a timely manner.
3. **User-friendly Interface:** The product should have a user-friendly interface that allows users to easily interact with the system, input data, configure settings, and view the results.
4. **Flexibility**: The product should be flexible and adaptable to different use cases and requirements. It should allow users to customize and configure the system, such as adjusting the algorithm parameters, incorporating domain-specific features, or integrating with other tools or systems.
5. **Integration**: The product should be designed to integrate with other existing systems or workflows, such as social media monitoring tools, news aggregators, or analytics platforms.
6. **Reliability and Security:** The product should be reliable and robust, able to handle errors, exceptions, and edge cases gracefully. It should also prioritize data security and privacy, ensuring that sensitive information, such as user data or tweet contents, are handled securely and protected against unauthorized access.

Overall, the product perspective for the "Fake Tweet Detection using Multinomial Naive Bayes Classifier Algorithm" project would be 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.

2.2 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.

2.3 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.

2.4 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. The choice of programming language, libraries, and hardware infrastructure can play a crucial role in achieving scalability and efficiency.
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.

Chapter 3

EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interface

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.

3.2 Hardware Interfaces

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

3.3 Software Interfaces

Tool: Pycharm 2016

Language: Python 3.6

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

3.4 Communication Interfaces

Fake Tweet Detection Using Multinomial Naive Bayes Classifier Algorithm is a web application. Hence, users can use this web application through browsers.

Chapter 4

FUNCTIONAL REQUIREMENTS

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

1. Input test case must not have compilation and runtime errors.
2. The application must not stop working when kept running for even a long time.
3. The application must function as expected for every set of test cases provided.
4. The application should generate the output for given input test case and input
5. parameters.
6. The application should generate on-demand services.

Chapter 5

OTHER REQUIREMENTS

5.1 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.

5.2 Design Constraints and Attributes

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. The choice of programming language, libraries, and hardware infrastructure can play a crucial role in achieving scalability and efficiency.
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.

5.3 Security Requirements

The system the has following main security requirements:

1. Users must provide all the mandatory data before accessing application features.
2. Only the developers should be able to access the database and remove spammers.
3. Database contents should not be altered in any way.
4. The contents uploaded by the user must be the same when the new user logs in.
5. Proper error messages should be displayed when the password is mismatched.