VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

PROJECT WORK PHASE - 2 DESIGN & IMPLEMENTATION REPORT

ON

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

Submitted by

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

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 SREENIVAS SHENOY with USN 4CB19CS087, Mr. P PADMAPRASAD SHENOY with USN 4CB19CS064, Mr. SUBRAMANYA A SHET with USN 4CB19CS105, Mr. SUHAS S KAMATH with USN 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 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

LIST OF FIGURES

CHAPTER 1

ABSTRACT 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.

1.1 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 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.

The architectural design of the proposed system is shown in figure 1.1. Here the tweets have been identified and pre-processed, and all of their features have been extracted. The dataset has been trained using the Multinomial Naïve Bayes Classifier Algorithm. As a result, the tweets can now be classified as either fake or genuine

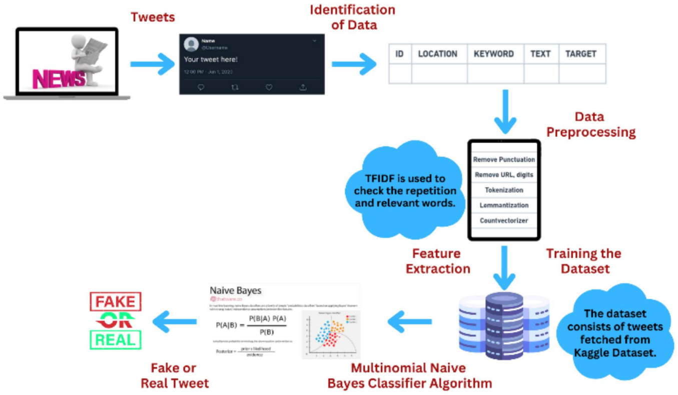

Fig 1.1 Architectural Design

CHAPTER 2

FUNCTIONAL DESIGN

2.1 Modular Design Diagram

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. Each of these individual components is then integrated (or assembled) together to form the final product. 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.

The product is a web-based application. It involves collecting Twitter data, preprocessing it to remove irrelevant information, training a machine learning classifier to classify the data, implementing the classifier into an application, using Matplotlib to generate visualizations of the data, and generating a graph to summarize the results of the analysis.

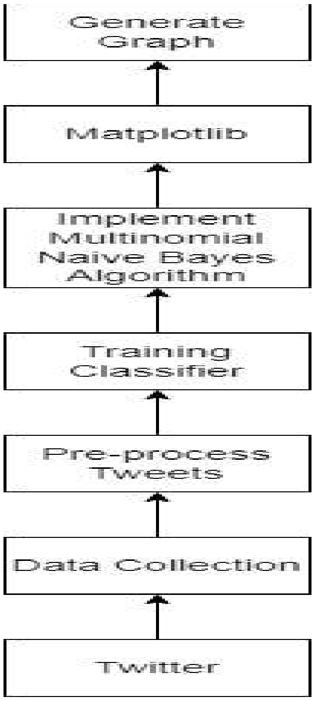

Fig 2.1 Modular Design Diagram of Proposed System

2.2 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

The process of detecting fake tweets involves several stages, including data collection, preprocessing, and analysis. In the data collection stage, tweets are obtained from Twitter using the Twitter API. These tweets are then preprocessed to remove stop words, special characters, and links. The preprocessing step is necessary to extract meaningful information from the tweet text.

Next, features are extracted from the preprocessed tweet text. These features include sentiment analysis, the frequency of specific words, and the number of hashtags used. These features are then used to train a machine learning algorithm, such as Naive Bayes, to classify the tweet as either real or fake.

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.

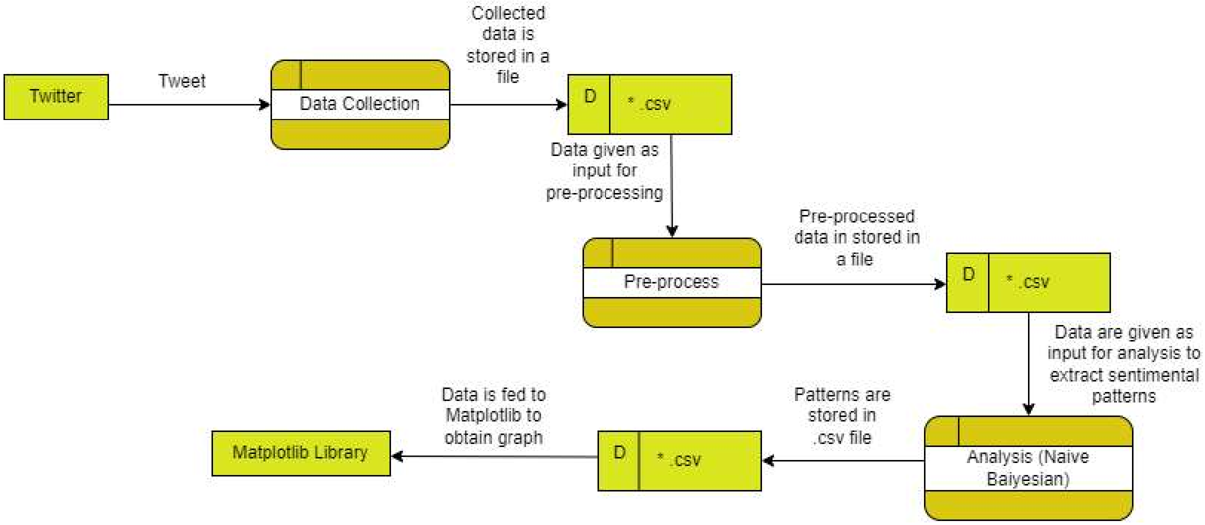
Twitter
Tweet
Data Collection
Collected data is stored in a file
D
* .csv
Data given as input for pre-processing
Pre-process
Pre-processed data in stored in a file
D
* .csv
Data are given as input for analysis to extract sentimental patterns
Analysis (Naive Baiyesian)
Patterns are stored in .csv file
D
* .csv
Data is fed to Matplotlib to obtain graph
Matplotlib Library

Fig 2.2 Data Flow Diagram

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.

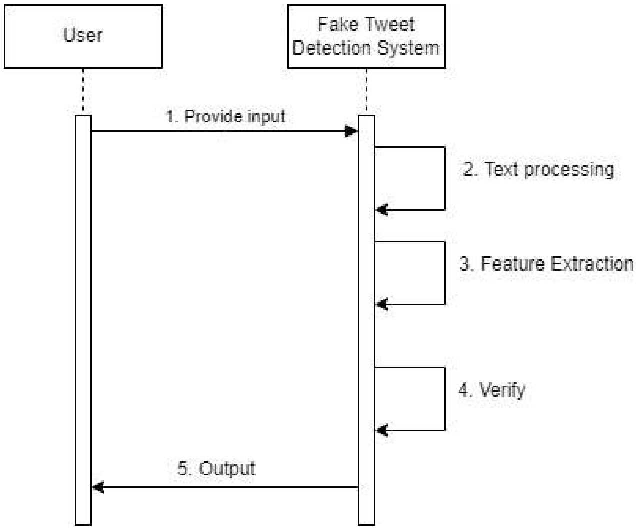

Fig 2.3 Sequence Diagram

2.4 Class Diagram

A class diagram is a type of UML (Unified Modelling Language) diagram that represents the structure and relationships of classes and objects in a system. It 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 might include removing irrelevant information, correcting spelling and grammar errors, and standardizing the text. The third stage, Classification, applies machine learning algorithms or other statistical techniques 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.

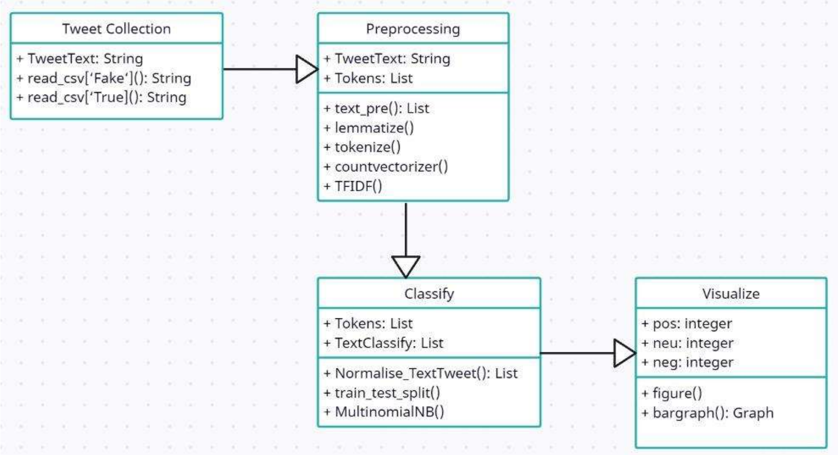

Fig 2.4 Class Diagram

CHAPTER 3

CONTROL FLOW DESIGN

3.1 Complete System Flow Diagram

The system flow diagram is one of the graphical representations of the flow of data in a system in software engineering. The diagram consists of several steps that identify where the input is coming to the system and output going out of the system. With the help of the diagram, it is possible to control the event decisions of the system and how data is flowing to the system. Therefore, the system flow diagram is basically a visual representation of data flow, excluding the minor parts and including the major parts of the system in a sequential manner.

The complete system flow design includes:

- Collect a dataset of tweets that are labelled as either fake or real.
- Pre-process the data by removing stop words, stemming and tokenizing the text.
- Select the most relevant features using feature selection techniques such as chi-square or mutual information.
- Split the dataset into training and testing sets.
- Train the Multinomial Naïve Bayes (MNB) classifier algorithm on the training set using the selected features.
- Evaluate the performance of the trained model on the testing set using metrics such as accuracy, precision, recall and F1-score.

3.2 Algorithm for Logic Implementation

The Algorithm for Implementation of Logic is given as follows:

- **Data collection:** The first step is to collect a dataset of tweets that are labelled as either fake or real. This dataset should be diverse, balanced, and representative of the problem domain.

- **Data pre-processing:** Once the data is collected, it needs to be pre-processed 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 pre-processed 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.

3.3 Activity Diagram for Use-Cases

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

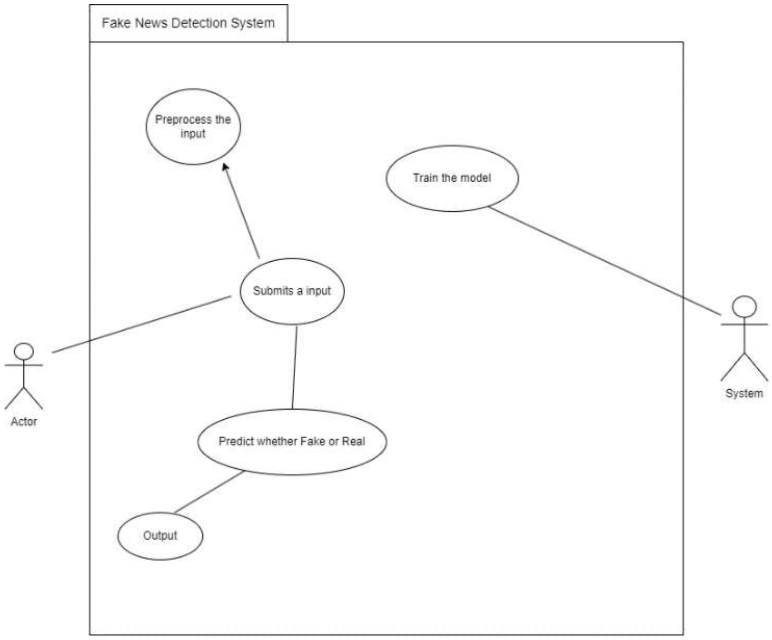

Fig 3.3 Activity Diagram

CHAPTER 4

PRESENTATION LAYER DESIGN

4.1 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.

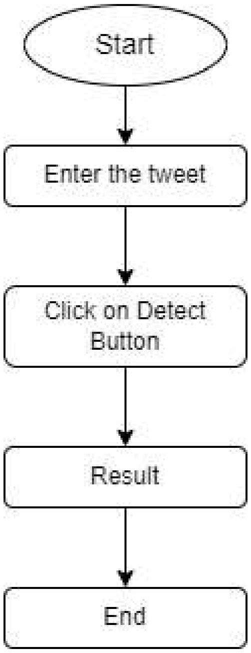

Fig 4.1 User Interface Flow Design