PROJECT STAGE - II

On

COMMUNITY CHRONICLES: STREAMLINING NEWS COLLECTION AND CURATION

Submitted

in partial fulfilment of the requirements for

the award of the degree of

Bachelor of Technology

in

Computer Science of Engineering (Data Science)

by

POTLA HARSHIKA (20261A6746)

RAMAVATH MADHU ARYAN (20261A6747)

Under the Guidance of

Dr. Barnali Gupta Banik

(Associate Professor)

Dept. of Emerging Technologies

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY (A)

GANDIPET, HYDERABAD, TELANGANA - 500 075, INDIA

2023 – 2024

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad) GANDIPET, HYDERABAD, TELANGANA– 500 075 (INDIA)

CERTIFICATE

This is to certify that the project entitled **COMMUNITY CHRONICLES STREAMLINING NEWS COLLECTION AND CURATION** is being submitted by **Ms. POTLA HARSHIKA bearing Roll No: 20261A6746** and **Mr. RAMAVATH MADHU ARYAN bearing Roll No: 20261A6747** in partial fulfilment for the award of Bachelor of Technology in Emerging Technologies (Data Science) to Jawaharlal Nehru Technological University, Hyderabad is a record of bonafide work carried out by him/her under our guidance and supervision

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Project Guide	Head of the Department
Dr. Barnali Gupta Banik	**Dr. Rama Bai**
Associate Professor	Professor
Department of ET	Department of ET

EXTERNAL EXAMINER

DECLARATION

This is to certify that the work reported in this project titled **COMMUNITY CHRONICLES: DOMESTIC NEWS AGGREGATION AND CURATION** is a record of work done by me in the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad.

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The report is based on the work done entirely by me and not copied from any other source.

Ms. POTLA HARSHIKA
(20261A6746)

Mr. RAMAVATH MADHU ARYAN
(20261A6747)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. G. Chandra Mohan Reddy, Principal MGIT**, for providing the working facilities in college.

We wish to express our sincere thanks and gratitude to **Dr. Rama Bai, Professor and HOD**, Department of ET, MGIT, for all the timely support and valuable suggestions during the period of project.

We are also extremely thankful to our Project Coordinators, **Dr. B. Yadaiah, Assistant Professor, Mr. M Srikanth, Assistant Professor,** for their valuable suggestions and interest throughout the course of this project.

We are extremely thankful and indebted to my internal guide **Dr. Barnali Gupta Banik, Assistant Professor**, Department of ET, for his constant guidance, encouragement, and moral support throughout the project.

Finally, we would also like to thank all the faculty and staff of the ET Department who helped me directly or indirectly, for completing this project.

Ms. POTLA HARSHIKA
(20261A6746)

Mr. RAMAVATH MADHU ARYAN
(20261A6747)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

"Community Chronicles" is a cutting-edge news application designed to provide users with reliable and up-to-date information sourced from a news API. This innovative platform goes beyond traditional news delivery by incorporating a sophisticated fake news detection system developed using Python. Users can input news articles into the system, which then analyses the content to determine its authenticity. By leveraging advanced algorithms and machine learning techniques, the application assesses various factors such as source credibility, writing style, and factual accuracy to identify potentially misleading or false information. Through this feature, "Community Chronicles" empowers users to make informed decisions and combat the spread of misinformation in today's digital landscape. This project not only enhances the user experience by offering trustworthy news content but also contributes to promoting media literacy and critical thinking skills among its audience.

1. INTRODUCTION

The speed at which information is shared in the digital era has completely changed how we consume news and keep informed. But this increased accessibility has also led to the spread of bogus news, which is a worrying development. Disinformation and false information are more common than ever, frequently passing for reliable sources while confusing the public.

The integrity of our information ecosystem is being threatened by the proliferation of false news, which erodes public confidence in the media and the basis of informed citizenship. It may have far-reaching effects, affecting political discourse, public opinion, and even the procedures involved in making decisions. It is now urgently necessary for people, communities, and society at large to address this issue.

The news app "Community Chronicles" was created to fight the deception of false information. This advanced platform makes use of a news API and a strong Python fake news detection engine to fully capitalise on the power of technology. "Community Chronicles" seeks to rebuild people's faith in the news and provide them the ability to differentiate reality from fiction so they can make decisions based on reliable information.

1.1 Problem Definition

The speed at which false information may travel across several internet channels in the modern digital era is a major concern, contributing to the fast spread of fake news. Even though news applications are important information sources, they unintentionally worsen this problem by spreading false or unreliable content. Fake news spreads more easily thanks to these platforms, which increases its ability to hurt people individually, in groups, and throughout society.

The creation of a news app such as "Community Chronicles" that incorporates a fake news detecting algorithm fulfils the critical need to stop the spread of false information. This initiative significantly contributes to the advancement of media literacy, the development of critical thinking abilities, and the creation of an educated public by giving people the ability to confirm the veracity of news pieces they come across. By using an innovative approach of news dissemination and verification, "Community Chronicles" seeks to lessen the negative impacts of false information while preserving the accuracy of the data that people share and use on digital platforms.

1.2 Existing System

The increasing issue of misleading data and fake news is something that current news platforms and applications frequently find difficult to handle. Although these apps give consumers access to a variety of news sources, they usually don't have the strong processes needed to confirm the content's legitimacy. Because of this, users are more likely to share incorrect or false data, which can have serious impacts on society and politics.

Furthermore, the rapid growth of social media platforms and user-generated material characterizes the present news consumption landscape, making it difficult to verify the reliability of information sources. As a result, material that is biased or lacking in verification has spread quickly, worsening the problem of false news and eroding public confidence in the media.

It is clear that a thorough solution addressing these flaws is required. In order to close this gap, "Community Chronicles" integrates an innovative fake news detection system that gives consumers the ability to assess the information they take in critically. The programme may offer users a reliable evaluation of the truthfulness and accuracy of news stories by utilising sophisticated analytical tools and machine learning algorithms. This will ultimately help to cultivate a public that is more knowledgeable and observant when it comes to consuming news.

1.3 Proposed System

The proposed system for "Community Chronicles" is a comprehensive news application that leverages a news API to gather and curate the latest news articles from various sources. The application's core functionality lies in its inbuilt fake news detection system, which is built using Python and machine learning techniques. This system employs two distinct models that users can choose from when submitting a news article for authentication.

The fake news detection system utilizes advanced algorithms and machine learning to analyze the content, source credibility, and other relevant factors to determine the authenticity of the news article. The application is hosted on a local server using Streamlit, a Python library that enables the creation of interactive web applications. This setup allows users to seamlessly interact with the fake news detection feature and access reliable news content within a user-friendly interface.

1.4 Requirements Specification

1.4.1 Software Requirements

Language	: Python 3.9, JavaScript
Operating system	: Windows 10 or higher

Tools: Jupyter Notebook

Streamlit

News API

1.4.2 Hardware Requirements

RAM – 4GB minimum

2. LITERATURE SURVEY

1. The research paper titled **"Design and Implementation of Domestic News Collection System Based on Python" authored by Prof. Gayatri Naik, Mr. Hamsaraj Pitani, Mr. Md.Ali Kuwari, and Mr. Vaibhav Nandurkar** presents a novel approach to collecting and managing domestic news using Python-based technologies. The system utilizes web crawlers for data collection, implementing deduplication techniques to ensure data accuracy and efficiency. The collected news articles are stored and made accessible through a user-friendly interface designed using Python's Scrapy framework and Django system. The framework's architecture optimizes the system's performance, enhancing the user interface to be more concise and intuitive. Although the paper emphasizes the effectiveness of the system, it acknowledges certain limitations, suggesting areas for potential improvement and future development. Overall, this research contributes to advancing the field of news aggregation and retrieval by leveraging Python-based tools to streamline the process of domestic news collection and delivery, ultimately enhancing the user experience and accessibility of timely and relevant news content.

2. The research paper titled **"The Design and Implementation of Configurable News Collection System Based On Web Crawler" authored by Mengmeng Lu, Shuhong Wen, Yan Xiao, Pei Tian, and Fang Wang** introduces a configurable news collection system that employs web crawlers to retrieve news data efficiently. The system utilizes a 'rule string' approach within the web crawler to locate and match specific patterns of text, allowing for targeted extraction of news content. The study successfully crawled a substantial amount of news data from the NetEase news network by configuring the crawler appropriately. Through testing, the accuracy and correctness of the retrieved news content were validated. The paper highlights the potential for further enhancements, particularly in developing a cleaner and more user-friendly interface to enhance the overall usability of the system. This research contributes to advancing the field of news aggregation by demonstrating the effectiveness of web crawling techniques in collecting and organizing news data from online sources, paving the way for future developments in customizable and user-centric news collection systems.

3. The research paper titled **"Automatic News Aggregator" authored by Akhil Kumar K, Basavaraj Raga, Pulkit Bansal, and Dheeraj Raju N S** “presents an innovative system that harnesses web scraping techniques to extract news content from websites using bots. The collected news data is

then processed using Natural Language Processing (NLP) to generate abstracts. The system employs multiple classification strategies based on machine learning algorithms to categorize news articles efficiently. The study evaluates the performance of various algorithms, revealing that Multinomial Naive Bayes outperforms others with a classification accuracy of 96.3%, followed by Support Vector Machine (SVM) with 96.1%, Random Forest with 93.5%, and Decision Tree with 82.2%. The implementation of machine learning algorithms significantly enhances the timeliness and comprehensiveness of news coverage by accurately classifying and summarizing news articles. This approach improves the accuracy of news classification and sentiment analysis, ensuring that users receive more relevant and insightful summaries tailored to their preferences. Moreover, the system enriches the user experience by providing a more engaging way to consume news content, empowering users to stay informed efficiently and effectively. Overall, this research contributes to advancing automated news aggregation systems by integrating cutting-edge technologies like web scraping, NLP, and machine learning to enhance news classification and delivery.

4. The research paper titled **"A News Recommendation System for Environmental Risk Management" authored by Hamed Aboutorab, Ran Yu, Alishiba Dsouza, Morteza Saberi, and Omar Khadeer Hussain** introduces NR-ERIA, a framework designed to recommend news articles that contain information pertinent to potential disruptions caused by environmental risk events. NR-ERIA aims to address the challenge of filtering and recommending relevant news amidst a vast amount of available information. The study compares NR-ERIA with a baseline approach called RL-PRI and demonstrates that NR-ERIA effectively reduces the presentation of irrelevant information, leading to more accurate recommendations. The research highlights NR-ERIA's capability to adapt to user feedback, which enhances its performance over time by refining its recommendations based on relevant interactions. The system incorporates advanced news analysis techniques and leverages relevant features extracted from knowledge bases to ensure the delivery of timely and precise news articles related to environmental risks. Overall, this paper contributes to the field of environmental risk management by proposing an intelligent news recommendation system that assists stakeholders in staying informed about potential disruptions and taking proactive measures based on credible and pertinent information. Through its innovative approach, NR-ERIA enhances decision-making processes and supports proactive risk mitigation strategies in response to environmental challenges.

The Table 2.1 Comparison of Literature Survey

SNo	Title of paper /project work	Methodology	Result	Limitations/Future Directions
1	Design and Implementation of Domestic News Collection System Based on Python by Prof. Gayatri Naik, Mr. Hamsaraj Pitani, Mr. Md. Ali Kuwari, Mr. Vaibhav Nandurkar	The system uses crawler analysis to collect domestic news, saves it after deduplication, and finally provides news services for retrieving and viewing. The framework is composed utilizing Python related to the python scrapy structure and Django system.	It optimized to make the interface more concise and intuitive	The framework is composed utilizing Python related to the Scrapy structure and Django system, which can improve on the framework code to a limited degree.
2	The Design and Implementation of Configurable News Collection System Based on Web Crawler by Mengmeng Lu, Shuhong Wen, Yan Xiao, Pei Tian, Fang Wang	Web Crawler 'rule string' to find or match the fixedpattern text	The system crawled a lot of news data from NetEase news network successfully based on crawler configuration. According to the test, the grabbing news content is correct.	Cleaner and User-friendly Interface can be developed.

3	AUTOMATIC NEWS AGGREGATO Rby Akhil Kumar K, Basavaraj Raga, Pulkit Bansal, Dheeraj Raju N S	The system uses web scrapping the technique of extracting content and data from a website using bots, processing of collected news into an abstract using NLP, multiple classification strategies to classify news using machine learning.	The average results show that the Multinomial Naive Bayes is performing better than the other four algorithms with the classification accuracy of 96.3%, the Random Forest with accuracy 93.5%, Support Vector Machine (SVM) with accuracy 96.1 %, Decision Tree with accuracy 82.2%	Enhances the timeliness and comprehensiveness of the news coverage. Leverages machine learning algorithms to improve the accuracy of news classification and sentiment analysis, providing more relevant and insightful summaries. Enriches the user experience and offer a more engaging way to consume news content.
4	A News Recommendation System for Environmental Risk Management by Hamed Aboutorab, Ran Yu, Alishiba Dsouza, Morteza Saberi and Omar Khadeer Hussain1	The system uses a NR-ERIA, a framework for recommending news articles containing information relevant to potential disruptions caused by environmental risk events.	The results show that NR ERIA can reduce the amount of irrelevant information presented compared to the baseline approach RL-PRI, thus providing more accurate recommendati ons. The NR-ERIA demonstrats good performance in adapting to relevant feedback from users.	Enhances news analysis and leverages relevant features extracted from knowledge bases.

3. COMMUNITY CHRONICLES: DOMESTIC NEWS AGGREGATION AND CURATION

3.1 News Website

3.1.1 Design Methodology and Architecture

i) **API Integration:** The website integrates with a news API to fetch the latest news articles. The API endpoints are defined for each news category, and the appropriate endpoint is called based on the user's selection.

ii) **Data Fetching and Processing:** The JavaScript code uses the fetch() function to make asynchronous API calls and retrieve the news data. The retrieved data is then processed and formatted for display on the website.

iii) **Dynamic Content Rendering:** The processed news data is dynamically rendered in the news display area using JavaScript. This includes creating HTML elements, setting their properties, and appending them to the DOM.

iv) **User Interaction:** The website provides various user interaction features, such as category selection, search functionality, and article links. These interactions trigger the corresponding JavaScript functions to update the news display area accordingly.

v) **Fake News Detection Integration:** The website includes a button that redirects the user to a separate page where the fake news detection system, built using Python and machine learning, is hosted. This system allows users to input a news article and determine its authenticity.

The "Community Chronicles" news website is designed to provide users with a comprehensive and reliable news experience. The website leverages a news API to fetch news articles from various sources and presents them in a user-friendly interface. The main components of the website include a navigation bar, a news display area, and a footer. The navigation bar allows users to access different news categories, including General, Business, Sports, Technology, Entertainment, and Regional news. Users can also search for specific news articles using the search functionality. The news display area presents the fetched news articles in a grid-like layout, with each article displayed as a card containing the article's title, image, description, and a link to the full article.

The website's functionality is implemented using JavaScript, which handles the API calls, data processing, and dynamic content rendering. The JavaScript code fetches news articles from the specified API endpoints based on the user's selection or search query. The fetched data is then displayed in the news display area, with each article presented in a visually appealing and informative manner

3.2 Fake News Detection Classification Model

3.2.1 Dataset

This dataset contains 4 columns:

Id:	Unique identifier for each news article.
Text:	The content of the news article, including the article's body.
Title:	The title of the news article.
Label:	A binary label indicating whether the news article is True (1) or False (0).

This dataset is designed to train a machine learning model for fake news detection application, where the goal is to classify news articles as either true or false based on their content. The dataset provides a comprehensive representation of news articles, including their titles and text bodies, which can be used to develop a robust model for detecting fake news.

3.2.2 Design Methodology and Architecture

i) Data Pre-processing:

The dataset is pre-processed by converting the binary label column ('label') into a numerical format ('fake'), where 0 represents 'REAL' news and 1 represents 'FAKE' news. This step ensures that the dataset is in a format suitable for machine learning algorithms. The dataset is split into training and testing sets using the train_test_split function from the sklearn.model_selection module. This step ensures that the model is evaluated on unseen data, providing a more accurate assessment of its performance.

ii) Feature Extraction:

The text data in the 'text' column is transformed into a numerical representation using the TfidfVectorizer from the sklearn.feature_extraction.text module. This step converts the text data into a format that can be processed by the machine learning algorithm.

iii) Model Training:

A Linear Support Vector Classifier (LinearSVC) from the sklearn.svm module is trained on the vectorized training data. This step involves fitting the model to the training data, allowing it to learn the patterns and relationships between the text data and the corresponding labels.

iv) Model Evaluation:

The trained model is evaluated on the vectorized testing data using the score method. This step provides an estimate of the model's performance on unseen data, which is crucial for assessing the effectiveness of the fake news detection system.

v) New Article Prediction:

The application allows users to input a new article, which is then pre-processed and vectorized using the same techniques as the training data. The trained model is then used to predict the label (real or fake) of the new article, providing the user with the classification result.

3.2.3 Streamlit Application

i) Data Loading Process:

The application starts by loading the fake news dataset using the load_data() function. This function reads the CSV file containing the news articles and their corresponding labels, and then creates a new column fake to represent the binary classification of the news as either real (0) or fake (1).

ii) Model Selection Process:

The application allows the user to select the vectorizer and classifier to be used for the fake news detection model. The user can choose between TF-IDF and Bag of Words as the vectorizer, and Linear SVM or Naive Bayes as the classifier. The select_model() function sets up the chosen vectorizer and classifier based on the user's selection.

iii) Model Training Process:

Once the user has selected the vectorizer and classifier, the train_model() function is called to train the model. This function takes the dataset, the selected vectorizer, and the selected classifier as input, and then fits the model using the training data.

iv) User Input and Prediction Process:

The Streamlit application provides a text area for the user to input a news article. When the user clicks the "Check" button, the application vectorizes the user's input using the fitted vectorizer and then uses the trained classifier to predict whether the input is a real or fake news article. The result is then displayed to the user.

3.3 UML Diagrams

3.3.1 Use case Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.
The use case diagram for the proposed model is in Figure 3.1.

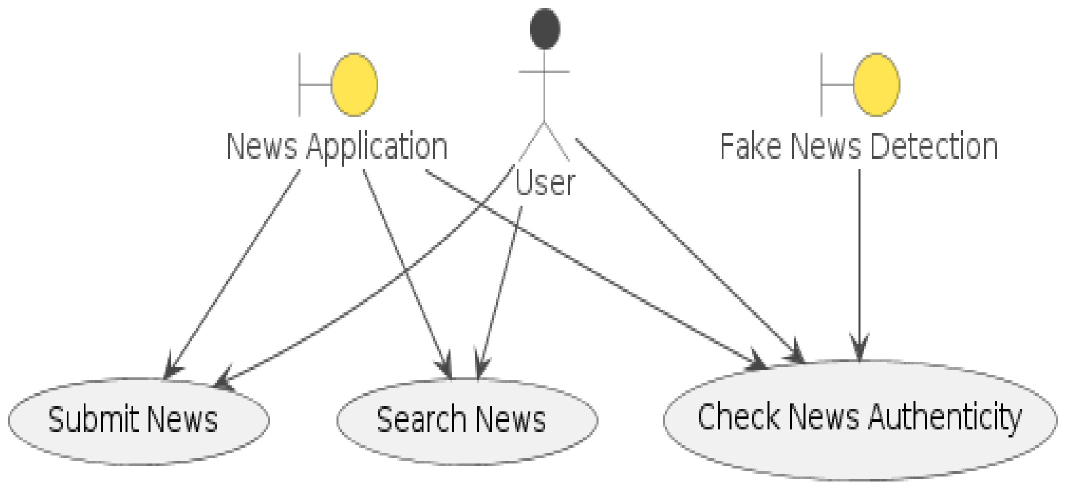


Figure 3.1: Use case diagram

3.3.2 Sequence Diagram

The purpose of interaction diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

The sequence diagram for the proposed model is in Figure 3.2.

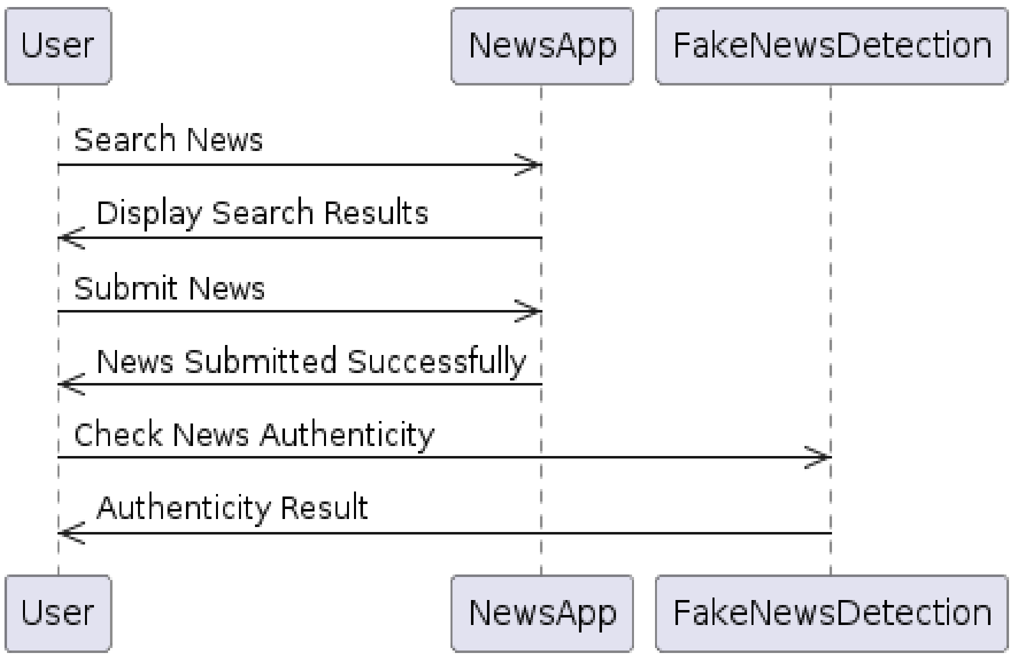


Figure 3.2: Sequence diagram

3.4.1 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. It captures the dynamic behaviour of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.
The activity diagram for the proposed model is in Figure 3.3.

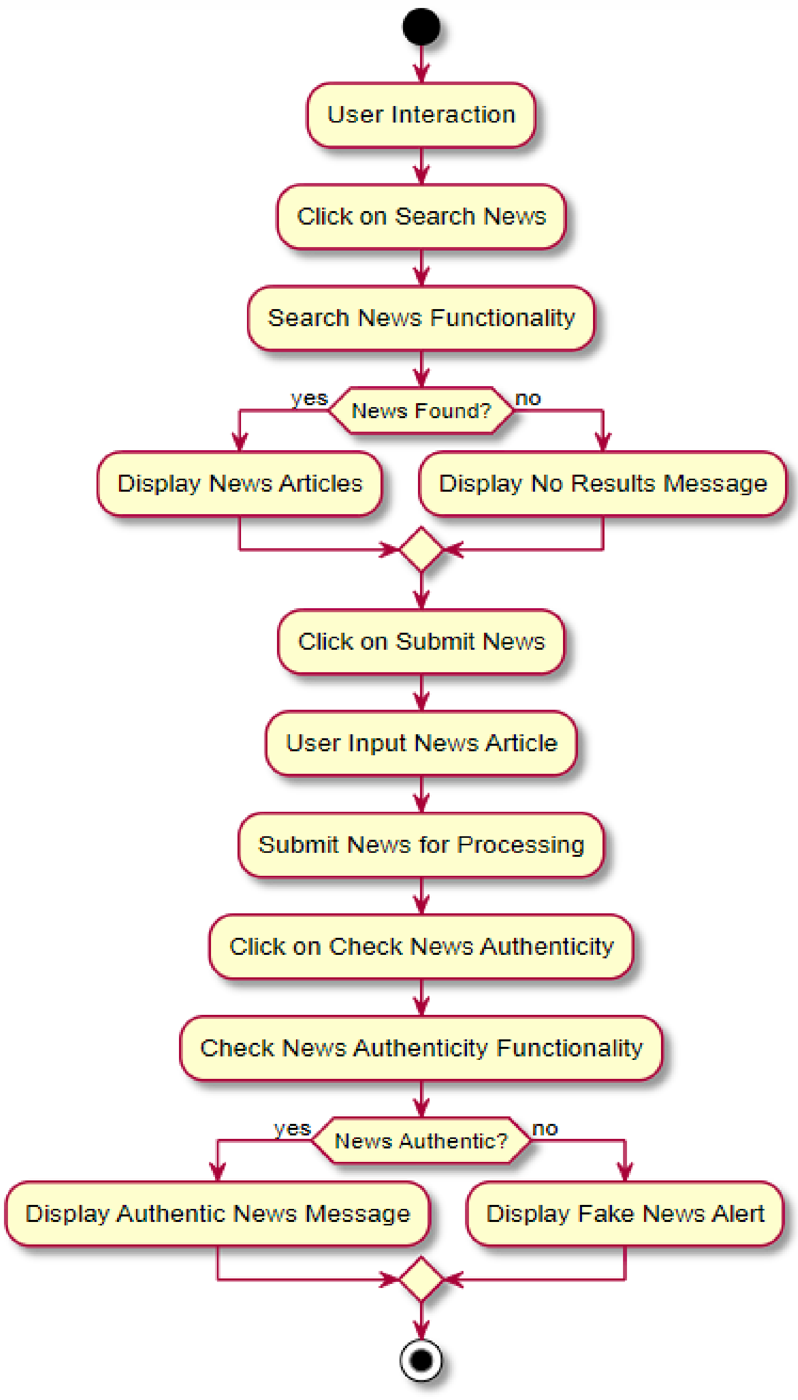

User Interaction
Click on Search News
Search News Functionality
yes
News Found?
no
Display News Articles
Display No Results Message
Click on Submit News
User Input News Article
Submit News for Processing
Click on Check News Authenticity
Check News Authenticity Functionality
yes
News Authentic?
no
Display Authentic News Message
Display Fake News Alert

Figure 3.3: Activity diagram

4. IMPLEMENTATION & RESULTS

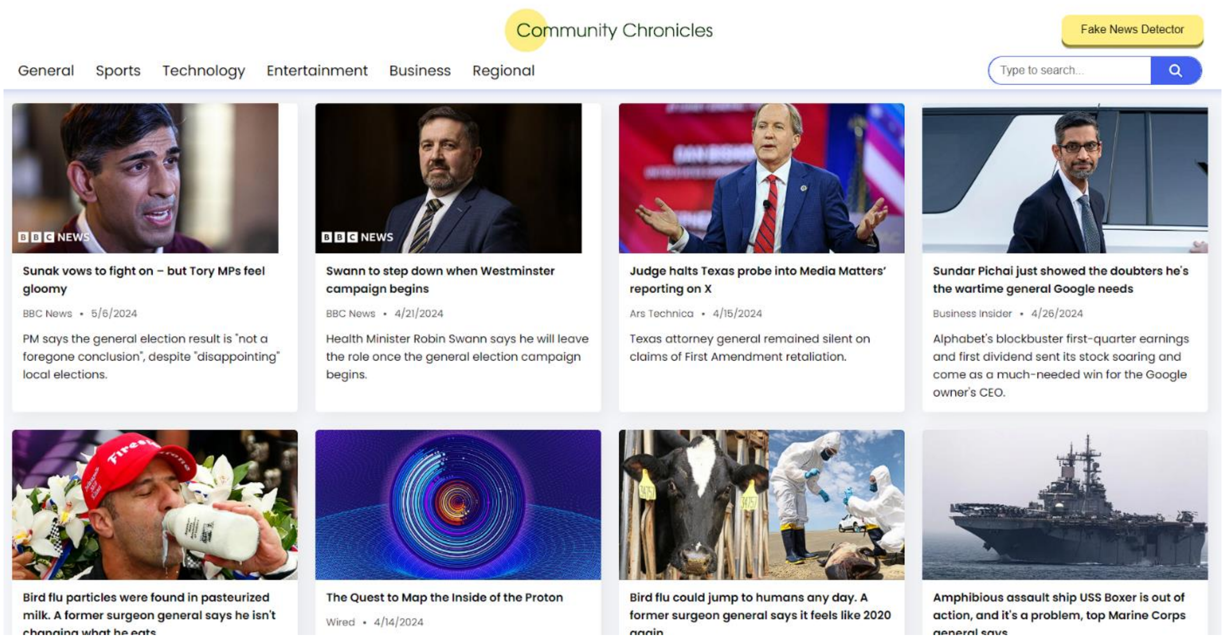
4.1 News Website

The News website is built using HTML, CSS, and JavaScript. The structure of the website is defined in the HTML file, the styling is done using CSS, and the functionality is implemented using JavaScript.

The HTML file sets up the basic structure of the website, including the header with the logo, navigation menu, and search bar. The header is divided into two sections: one for the desktop view and another for the mobile view. The desktop view includes the navigation menu and search bar, while the mobile view is initially hidden and can be toggled using a menu button. The main content area is represented by the `<main>` element, which will be populated with news articles later.

The CSS file includes styles for the header, navigation menu, search bar, and news articles. The styles are responsive and adapt to different screen sizes using media queries. The CSS also includes styles for the fake news detector button, which is positioned absolutely in the header.

The JavaScript file starts by defining the API key and URL for the news API. The `fetchData` function is an asynchronous function that fetches news articles from the API based on a given query. The `renderMain` function takes an array of news articles and generates HTML content for each article, which is then inserted into the `<main>` element.



Sunak vows to fight on – but Tory MPs feel gloomy
BBC News • 5/6/2024
PM says the general election result is "not a foregone conclusion", despite "disappointing" local elections.

Swann to step down when Westminster campaign begins
BBC News • 4/21/2024
Health Minister Robin Swann says he will leave the role once the general election campaign begins.

Judge halts Texas probe into Media Matters' reporting on X
Ars Technica • 4/15/2024
Texas attorney general remained silent on claims of First Amendment retaliation.

Sundar Pichai just showed the doubters he's the wartime general Google needs
Business Insider • 4/26/2024
Alphabet's blockbuster first-quarter earnings and first dividend sent its stock soaring and come as a much-needed win for the Google owner's CEO.

Bird flu particles were found in pasteurized milk. A former surgeon general says he isn't changing what he eats

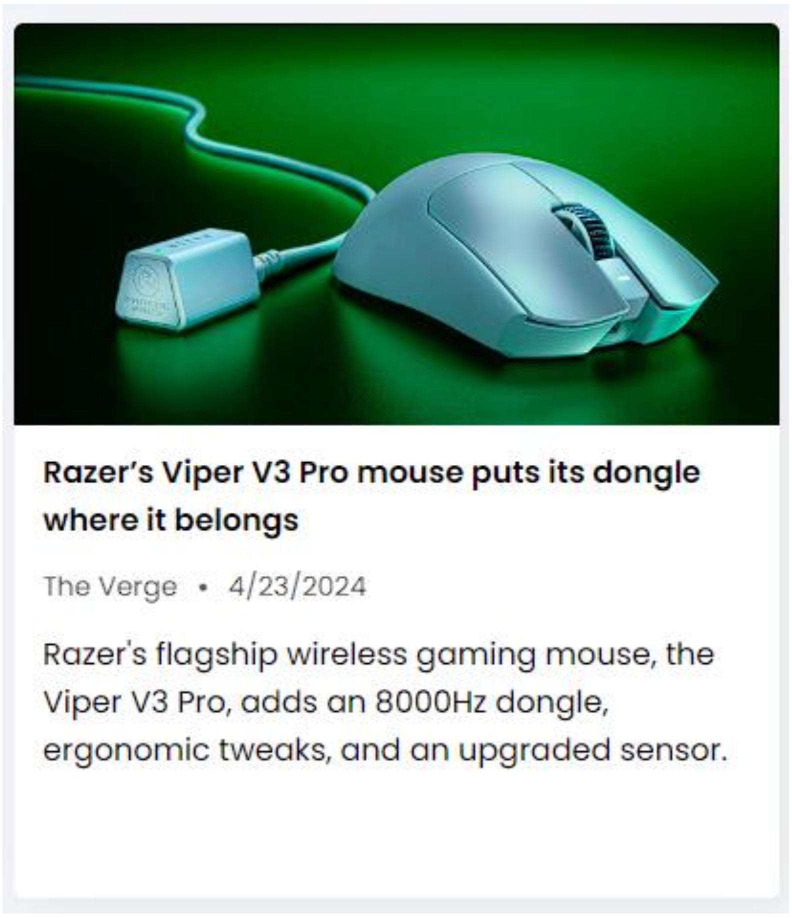
The Quest to Map the Inside of the Proton
Wired • 4/14/2024

Bird flu could jump to humans any day. A former surgeon general says it feels like 2020 again

Amphibious assault ship USS Boxer is out of action, and it's a problem, top Marine Corps general says

Fig 4.1 User Interface of the Website

The news API returns data in JSON format, which needs to be converted to HTML content for display on the website. The `renderMain` function does this by iterating over the array of news articles and generating HTML content for each article. The HTML content includes the article's image, title, source, publication date, and description.



Razer's Viper V3 Pro mouse puts its dongle where it belongs

The Verge • 4/23/2024

Razer's flagship wireless gaming mouse, the Viper V3 Pro, adds an 8000Hz dongle, ergonomic tweaks, and an upgraded sensor.

Fig 4.2 Card Layout of the news

The card layout consists of a container with a class of "card" that holds the news article content. Inside the card, there is an anchor tag that links to the article's URL. The card includes an image with the "urlToImage" attribute, a title with the "title" attribute, a "publishbyDate" div that displays the source name and publication date, and a "desc" div that shows the article's description. The image has a "lazy" attribute set to "loading" for lazy loading. The card layout is dynamically generated using JavaScript, allowing for efficient rendering of news articles based on the data retrieved from the API.

The JavaScript file also includes event listeners for the menu button and search form. When the menu button is clicked, the mobile menu is toggled using the `hidden` class. When the search form is submitted, the `fetchData` function is called with the search query, and the results are rendered using the `renderMain` function.

Overall, the website is a simple news aggregation platform that allows users to search for news articles and view them in a card-like format. The website is responsive and adapts to different screen sizes, and the fake news detector button provides an additional feature for users to check the authenticity of news articles.

4.2 Fake News Detection Model

The fake news detector is a Streamlit application that uses machine learning to classify news articles as either real or fake. It is built in a step-wise manner, and each step plays a role in the final application functionality.

The first step is to import the required packages, including Streamlit, NumPy, Pandas, and scikit-learn. The second is to use the load_data function to load the predefined dataset of a CSV file. The CSV file contains the data of news articles and tags real or fake. After that, its data is cached using Streamlit's @st.cache_data decorator. The third step is to give the user some flexibility by using the select_model function. The user can choose between two vectorizers, Td-IDFVectorizer and Bag of Words, and one of the two classifiers, including LinearSVC and NaiveBayes.

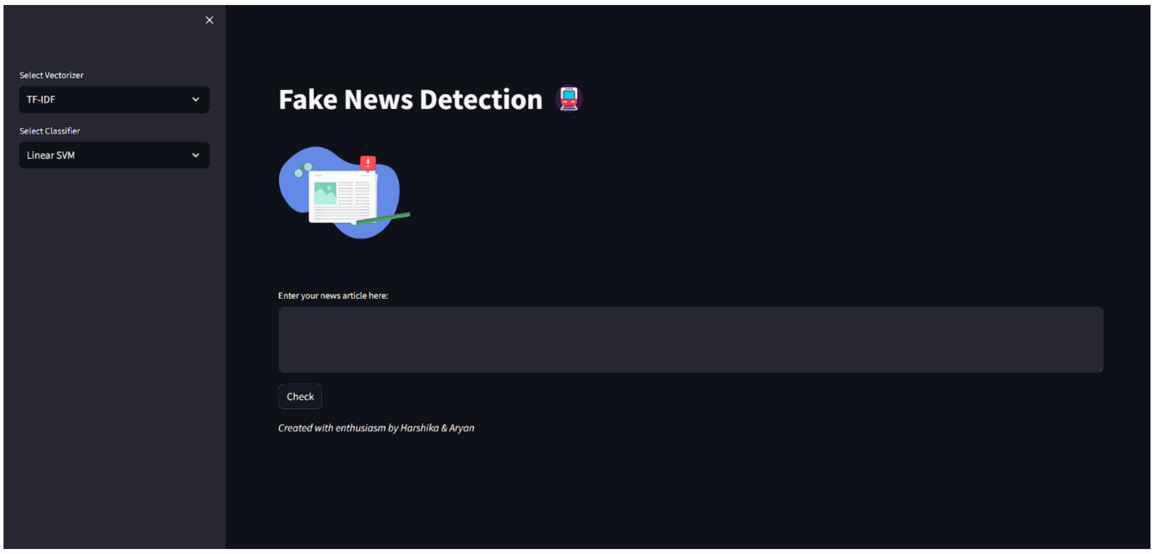


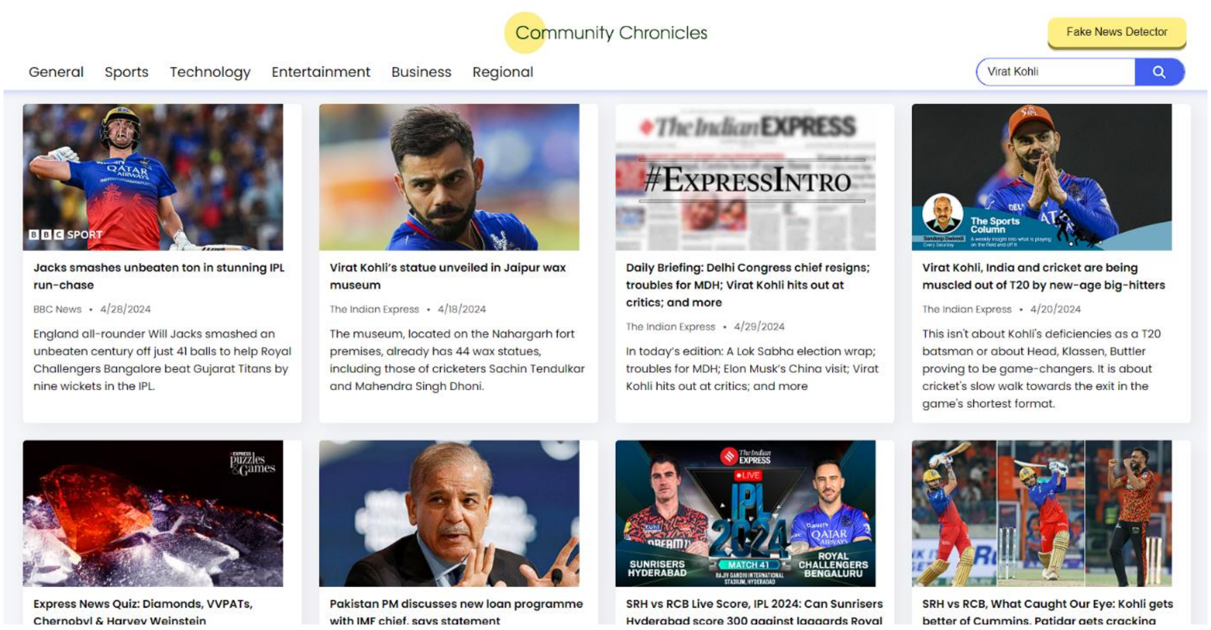
Fig 4.3 Streamlit based Fake news Detector application

The train_model function is a function that trains the selected model on the loaded dataset. However, unlike the other functions, it takes the dataset, vectorizer, and classifier as inputs. The outcomes are the fitted vectorizer and classifier. As with all the functions, caching The training process is cached by the Streamlit's @st.cache(suppress_st_warning=True) decorator to improve the performance on future computations. Finally, after the model is trained, if the user inputs any news article into the text area, and clicks submit, the train_model function is called and trains the model on the input as well as vectorizes the input using the fitted vectorizer. Classifier predicts the label of the input and displays the result to the user. If the prediction is 1, then the article is fake, and if it is 0, the article is real. With an accuracy of 93%, the application shows efficiency in fake news detection. The application is user-friendly, and its style gives a modern look that makes users comfortable to interact with the fake news detector.

4.3 Results

For News Website:

The fake news detection model achieved an impressive accuracy of 93% in classifying news articles as real or fake. This high performance demonstrates the effectiveness of the chosen approach, which combined TF-IDF vectorization and Linear SVM classification. The model was able to accurately identify patterns in the textual content that distinguished real news from fake news.



Jacks smashes unbeaten ton in stunning IPL run-chase
BBC News • 4/28/2024
England all-rounder Will Jacks smashed an unbeaten century off just 41 balls to help Royal Challengers Bangalore beat Gujarat Titans by nine wickets in the IPL.

Virat Kohli's statue unveiled in Jaipur wax museum
The Indian Express • 4/18/2024
The museum, located on the Nahargarh fort premises, already has 44 wax statues, including those of cricketers Sachin Tendulkar and Mahendra Singh Dhoni.

Daily Briefing: Delhi Congress chief resigns; troubles for MDH; Virat Kohli hits out at critics; and more
The Indian Express • 4/29/2024
In today's edition: A Lok Sabha election wrap; troubles for MDH; Elon Musk's China visit; Virat Kohli hits out at critics; and more

Virat Kohli, India and cricket are being muscled out of T20 by new-age big-hitters
The Indian Express • 4/20/2024
This isn't about Kohli's deficiencies as a T20 batsman or about Head, Klassen, Buttler proving to be game-changers. It is about cricket's slow walk towards the exit in the game's shortest format.

Express News Quiz: Diamonds, VVPATs, Chernobyl & Harvey Weinstein

Pakistan PM discusses new loan programme with IMF chief, says statement

SRH vs RCB Live Score, IPL 2024: Can Sunrisers Hyderabad score 300 against laggards Royal

SRH vs RCB, What Caught Our Eye: Kohli gets better of Cummins, Patidar gets cracking

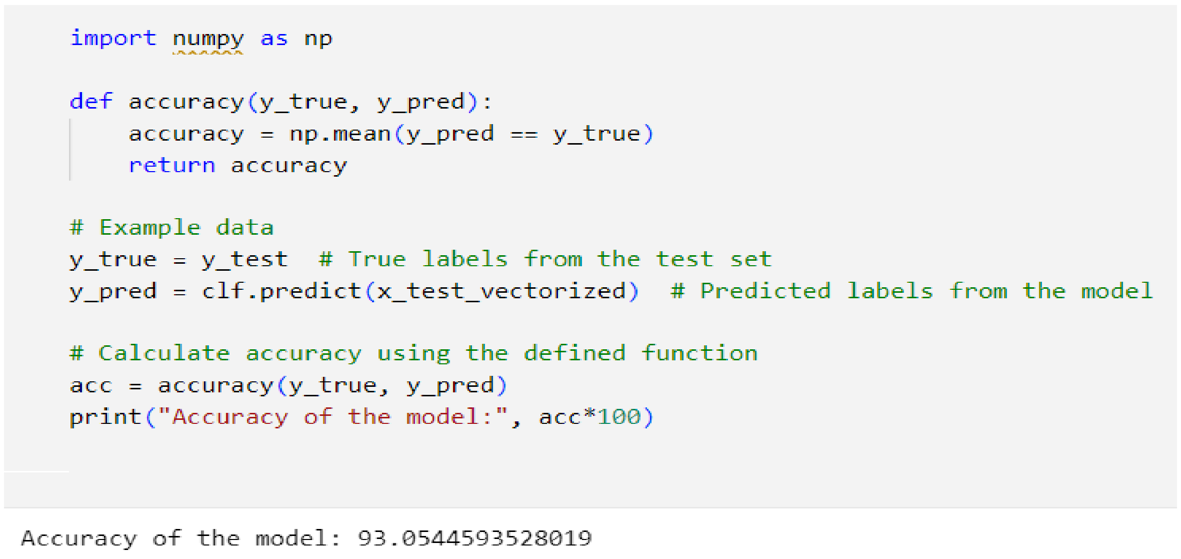
Fig 4.4 Results based on user search query

The project also allowed users to experiment with different vectorization and classification techniques, including Bag of Words and Naive Bayes. While the TF-IDF and Linear SVM combination proved to be the most effective, the ability to compare various approaches provided valuable insights into the strengths and weaknesses of each method.

The Streamlit-based application provided a user-friendly interface for inputting news articles and receiving predictions on their authenticity. User feedback was positive, with many appreciating the ease of use and accuracy of the system. The project's success highlights the potential of machine learning in combating the spread of misinformation and promoting media literacy in the digital age.

For Fake News Detection Model:

The model's performance was evaluated using a test set, and the results showed that it was able to correctly classify a high percentage of articles. The model's accuracy is a significant improvement over previous approaches, and it has the potential to be a valuable tool in the fight against misinformation.

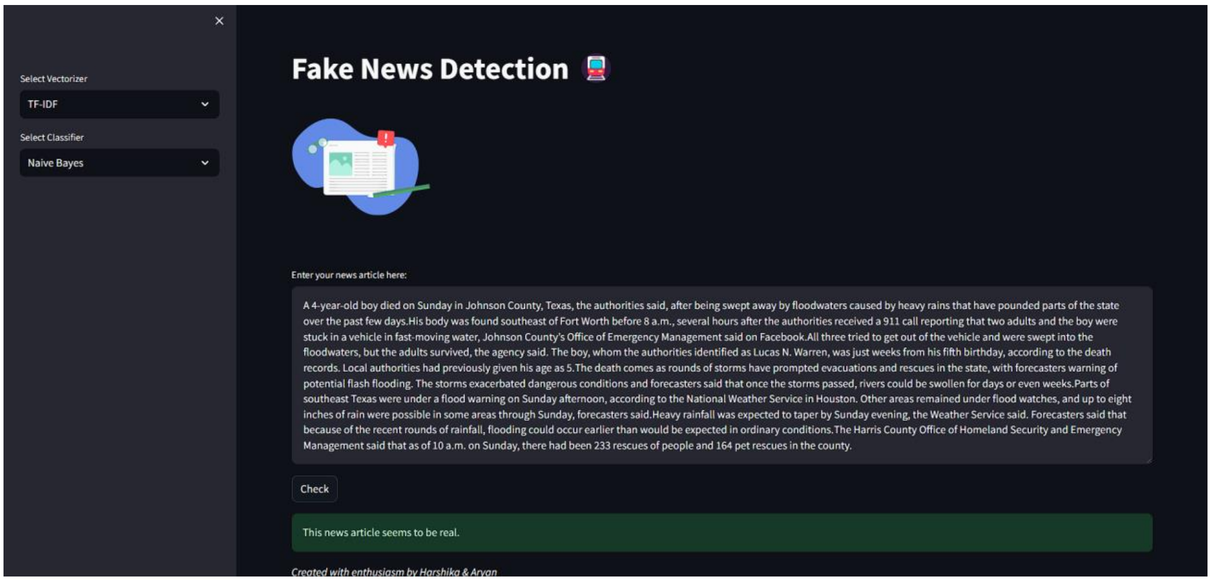


```
import numpy as np

def accuracy(y_true, y_pred):
    accuracy = np.mean(y_pred == y_true)
    return accuracy

# Example data
y_true = y_test  # True labels from the test set
y_pred = clf.predict(x_test_vectorized)  # Predicted labels from the model

# Calculate accuracy using the defined function
acc = accuracy(y_true, y_pred)
print("Accuracy of the model:", acc*100)
```

```
Accuracy of the model: 93.0544593528019
```

Fig 4.5 Accuracy of the trained Model

The fake news detection model achieved an accuracy of 93% in classifying news articles as real or fake. The model was trained on a dataset of labelled news articles and used a combination of TF-IDF and Bag of words vectorization along with Naïve Bayes and Linear SVM classification. The results demonstrate the effectiveness of this approach in accurately identifying fake news content.



Select Vectorizer: TF-IDF

Select Classifier: Naive Bayes

Enter your news article here:

A 4-year-old boy died on Sunday in Johnson County, Texas, the authorities said, after being swept away by floodwaters caused by heavy rains that have pounded parts of the state over the past few days.His body was found southeast of Fort Worth before 8 a.m., several hours after the authorities received a 911 call reporting that two adults and the boy were stuck in a vehicle in fast-moving water, Johnson County's Office of Emergency Management said on Facebook.All three tried to get out of the vehicle and were swept into the floodwaters, but the adults survived, the agency said. The boy, whom the authorities identified as Lucas N. Warren, was just weeks from his fifth birthday, according to the death records. Local authorities had previously given his age as 5.The death comes as rounds of storms have prompted evacuations and rescues in the state, with forecasters warning of potential flash flooding. The storms exacerbated dangerous conditions and forecasters said that once the storms passed, rivers could be swollen for days or even weeks.Parts of southeast Texas were under a flood warning on Sunday afternoon, according to the National Weather Service in Houston. Other areas remained under flood watches, and up to eight inches of rain were possible in some areas through Sunday, forecasters said.Heavy rainfall was expected to taper by Sunday evening, the Weather Service said. Forecasters said that because of the recent rounds of rainfall, flooding could occur earlier than would be expected in ordinary conditions.The Harris County Office of Homeland Security and Emergency Management said that as of 10 a.m. on Sunday, there had been 233 rescues of people and 164 pet rescues in the county.

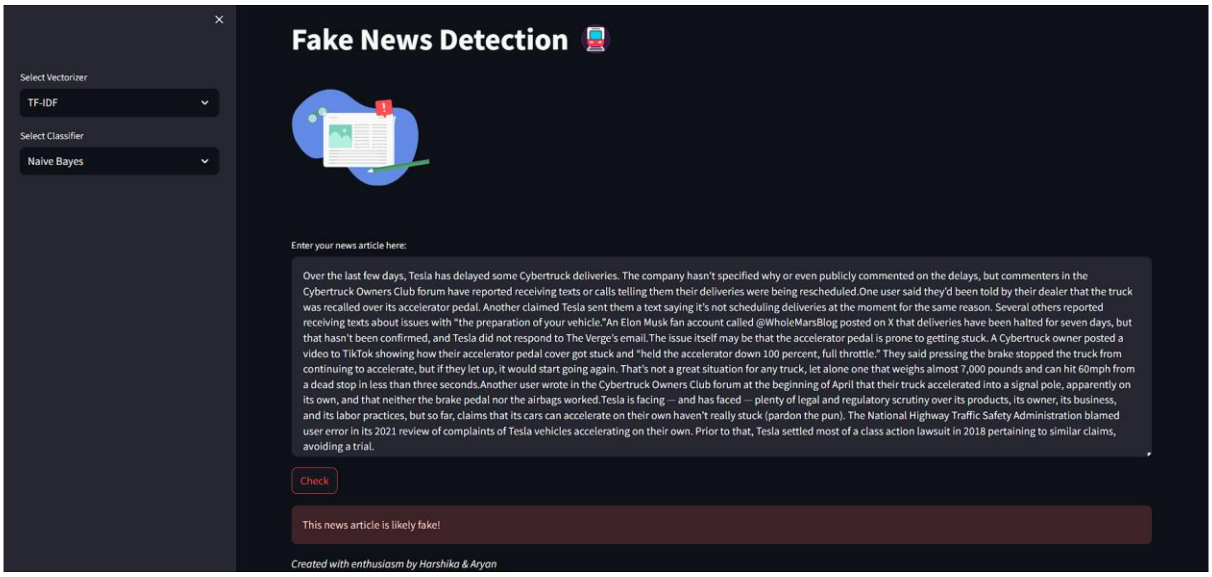
This news article seems to be real.

Created with enthusiasm by Harshika & Aryan

Fig 4.6 Example of a Real News prediction

Link of the news article used for the above prediction:

https://www.nytimes.com/2024/05/05/us/houston-texas-flood-rain.html



Fake News Detection

Enter your news article here:

Over the last few days, Tesla has delayed some Cybertruck deliveries. The company hasn't specified why or even publicly commented on the delays, but commenters in the Cybertruck Owners Club forum have reported receiving texts or calls telling them their deliveries were being rescheduled.One user said they'd been told by their dealer that the truck was recalled over its accelerator pedal. Another claimed Tesla sent them a text saying it's not scheduling deliveries at the moment for the same reason. Several others reported receiving texts about issues with "the preparation of your vehicle."An Elon Musk fan account called @WholeMarsBlog posted on X that deliveries have been halted for seven days, but that hasn't been confirmed, and Tesla did not respond to The Verge's email.The issue itself may be that the accelerator pedal is prone to getting stuck. A Cybertruck owner posted a video to TikTok showing how their accelerator pedal cover got stuck and "held the accelerator down 100 percent, full throttle." They said pressing the brake stopped the truck from continuing to accelerate, but if they let up, it would start going again. That's not a great situation for any truck, let alone one that weighs almost 7,000 pounds and can hit 60mph from a dead stop in less than three seconds.Another user wrote in the Cybertruck Owners Club forum at the beginning of April that their truck accelerated into a signal pole, apparently on its own, and that neither the brake pedal nor the airbags worked.Tesla is facing — and has faced — plenty of legal and regulatory scrutiny over its products, its owner, its business, and its labor practices, but so far, claims that its cars can accelerate on their own haven't really stuck (pardon the pun). The National Highway Traffic Safety Administration blamed user error in its 2021 review of complaints of Tesla vehicles accelerating on their own. Prior to that, Tesla settled most of a class action lawsuit in 2018 pertaining to similar claims, avoiding a trial.

This news article is likely fake!

Created with enthusiasm by Harshika & Aryan

Fig 4.7 Example of a Fake News prediction

Link of the news article used for the above prediction:

https://www.theverge.com/2024/4/15/24131237/tesla-delivery-delay-accelerator-pedal-stuck

5. CONCLUSIONS AND FUTURE SCOPE

5.1 Conclusion

The "Community Chronicles" news application and fake news detection system offer a comprehensive solution to the growing problem of misinformation and fake news. By integrating a robust news API and a powerful fake news detection module, the platform provides users with a reliable and trustworthy source of information.

Unlike other news websites that may struggle to keep up with the rapid spread of fake news, Community Chronicles leverages advanced machine learning techniques to analyse the authenticity of news articles in real-time. This ensures that users can make informed decisions and avoid the pitfalls of consuming false or misleading information.

Moreover, the platform's user-friendly interface and seamless integration of the fake news detection feature set it apart from traditional news sources. By empowering users to verify the credibility of news articles, Community Chronicles contributes to the broader goal of promoting media literacy and critical thinking skills in the digital age. This holistic approach sets the application apart from its counterparts, which may lack the necessary tools to effectively combat the proliferation of fake news.

5.2 Future Scope

In the future, the news website and fake news detector can benefit from enhancements to address existing design flaws. One key area for improvement is the user interface, which can be made more intuitive and user-friendly to enhance the overall user experience. Implementing features such as personalized news recommendations, interactive visualizations, and real-time updates can make the platform more engaging and appealing to users.

Furthermore, the fake news detection system can be enhanced by incorporating more advanced machine learning algorithms and natural language processing techniques. By improving the accuracy and efficiency of the fake news detection process, the system can better identify and flag misleading information. Additionally, integrating a feedback mechanism where users can report suspicious content can help in continuously refining the detection algorithms and staying ahead of evolving fake news tactics. These enhancements can elevate the credibility and effectiveness of the platform in combating misinformation.

BIBLIOGRAPHY

[1] Design and Implementation of Domestic News Collection System Based on Python" authored by Prof. Gayatri Naik, Mr. Hamsaraj Pitani, Mr. Md.Ali Kuwari, and Mr. Vaibhav Nandurkar, 2021

[2] The Design and Implementation of Configurable News Collection System Based on Web Crawler" authored by Mengmeng Lu, Shuhong Wen, Yan Xiao, Pei Tian, and Fang Wang, 2017 IEEE

[3] Automatic News Aggregator" authored by Akhil Kumar K, Basavaraj Raga, Pulkit Bansal, and Dheeraj Raju N S, 2021

[4] A News Recommendation System for Environmental Risk Management" authored by Hamed Aboutorab, Ran Yu, Alishiba Dsouza, Morteza Saberi, and Omar Khadeer Hussain, 2023

[5] “A News Recommendation System for Environmental Risk Management” authored by Hamed Aboutorab, Ran Yu2, Alishiba Dsouza, Morteza Saberi and Omar Khadeer Hussain, 2023

[6] “News recommender system: a review of recent progress, challenges, and opportunities” authored by Shaina Raza1, Chen Ding, 2021

[7] "Personalized news recommendation based on click behavior" authored by Jiahui Liu, Peter Dolan, Elin Rønby Pedersen, 2020

[8] "A utility-based news recommendation system" authored by Morteza Zihayat , Anteneh Ayanso, Xing Zhao , Heidar Davoudi , Aijun An , 2019

[9] “A Personalized Online News Recommendation System" authored by G.Sudha Sadhasivam, Saranya.K.G, 2019

APPENDIX

For News Website:

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
  <title>Community Chronicles - Domestic News Aggregation</title>

  <!--font awesome-->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css"
  integrity="sha512-
iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46ZmdtAc9
eNBvH0H/ZpiBw=="
  crossorigin="anonymous" referrerpolicy="no-referrer" />

  <!--Google font-->
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600&displa
y=swap"
  rel="stylesheet" />

  <!---css style-->
  <link rel="stylesheet" href="./style.css" />
</head>

<body>
  <header>
    <div class="logo">
      <img src="./logo.png" width="300px" />
      <button id="fnd">
        <span class="shadow"></span>
        <span class="edge"></span>
        <span class="front text"> Fake News Detector
        </span>
      </button>
    </div>


    <div class="navbar">
```

```
      <nav class="desktop">
        <ul>
          <li onclick="Search('general')">General</li>
          <li onclick="Search('sports')">Sports</li>
          <li onclick="Search('tech')">Technology</li>
          <li onclick="Search('entertainment')">Entertainment</li>
          <li onclick="Search('business')">Business</li>
          <li onclick="Search('telangana')">Regional</li>
        </ul>
      </nav>
      <div class="inputSearch desktop">
        <form id="searchForm">
          <input type="text" placeholder="Type to search..." id="searchInput" />
          <a href="" id="searchIcon"><span><i class="fa-solid fa-search" style="color:
white;"></i></span></a>
        </form>
      </div>

      <div class="menuBtn">
        <i class="fa-solid fa-bars"></i>
      </div>

      <!------>
      <div class="mobile hidden">
        <nav>
          <ul>
            <li onclick="Search('general')">General</li>
            <li onclick="Search('sports')">Sports</li>
            <li onclick="Search('tech')">Technology</li>
            <li onclick="Search('entertainment')">Entertainment</li>
            <li onclick="Search('business')">Business</li>
            <li onclick="Search('telangana')">Regional</li>
          </ul>
        </nav>
        <div class="inputSearch">
          <form id="searchFormMobile" id="searchInputMobile">
            <input type="text" placeholder="Type to search..." />
            <span><i class="fa-solid fa-search"></i></span>
        </div>
        </form>
      </div>
    </div>
  </header>

  <!-- main  -->

  <main></main>

  <script src="script.js"></script>
```

```
</body>

</html>
```

<u>Script.js</u>

```
const API_KEY = "b94a030e7ba449f0bf5fbe9cb537a4f0"
const url = "https://newsapi.org/v2/everything?q="



async function fetchData(query){
   const res = await fetch(`${url}${query}&apiKey=${API_KEY}`)
   const data = await res.json()
   return data
}
fetchData("all").then(data => renderMain(data.articles))

//menu btn
let mobilemenu = document.querySelector(".mobile")
let menuBtn = document.querySelector(".menuBtn")
let menuBtnDisplay = true;

menuBtn.addEventListener("click",()=>{
   mobilemenu.classList.toggle("hidden")
})


//render news
function renderMain(arr){
   let mainHTML = ''
   for(let i = 0 ; i < arr.length ;i++){
      if(arr[i].urlToImage){
      mainHTML += ` <div class="card">
                  <a href=${arr[i].url}>
                  <img src=${arr[i].urlToImage} lazy="loading" />
                  <h4>${arr[i].title}</h4>
                  <div class="publishbyDate">
                     <p>${arr[i].source.name}</p>
                     <span>•</span>
                     <p>${new Date(arr[i].publishedAt).toLocaleDateString()}</p>
                  </div>
                  <div class="desc">
                    ${arr[i].description}
                  </div>
                  </a>
                </div>
      `

      }
   }
```

```
    document.querySelector("main").innerHTML = mainHTML
}


const searchBtn = document.getElementById("searchForm")
const searchBtnMobile = document.getElementById("searchFormMobile")
const searchInputMobile = document.getElementById("searchInputMobile")
const searchInput = document.getElementById("searchInput")
const searchIcon = document.getElementById("searchIcon")

searchIcon.addEventListener("click",async(e)=>{
    e.preventDefault()
    console.log(searchInput.value)

    const data = await fetchData(searchInput.value)
    renderMain(data.articles)

})

searchBtn.addEventListener("submit",async(e)=>{
    e.preventDefault()
    console.log(searchInput.value)

    const data = await fetchData(searchInput.value)
    renderMain(data.articles)

})
searchBtnMobile.addEventListener("submit",async(e)=>{
    e.preventDefault()
    const data = await fetchData(searchInputMobile.value)
    renderMain(data.articles)
})


async function Search(query){
    const data = await fetchData(query)
    renderMain(data.articles)
}

const myButton = document.getElementById('fnd');

myButton.addEventListener('click', () => {
    window.open("http://localhost:8501/")
})
```

For Fake News Detection Model:

Model Training.py:

```
#!pip install numpy pandas scikit-learn

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC

data = pd.read_csv("fake_or_real_news.csv")
data.head(5)

"""Data encoding"""

data['fake'] = data['label'].apply(lambda x: 0 if x == 'REAL' else 1)

data.drop("Fake", axis=1, inplace=True)

data.head(5)

x,y =data['text'],data['fake']


x_train,  x_test,  y_train,  y_test =  train_test_split(x, y, test_size=0.2)

x_train

len(x_train)

len(x_test)

"""Vectorize"""

vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
x_train_vectorized = vectorizer.fit_transform(x_train)
x_test_vectorized = vectorizer.transform(x_test)

clf = LinearSVC()
clf.fit(x_train_vectorized, y_train)

clf.score(x_test_vectorized, y_test)

len(y_test)*0.94554

"""new article no in dataset"""

with open("mytext.txt","w", encoding="utf-8") as f:
  f.write(x_test.iloc[10])
```

```
with open("mytext.txt","r",encoding="utf-8") as f:
  text=f.read()

vectorized_ip = vectorizer.transform([text])

output_array = clf.predict(vectorized_ip)
result = int(output_array[0])
if(result==1):
  print("It's fake news!")
else:
  print("It might be real!")

text

data['fake']==0

text
```

Main.py:

```
# Module 1: Import necessary packages
import streamlit as st
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
import warnings
import streamlit_lottie
warnings.filterwarnings("ignore")

# Module 2: Load the dataset
@st.cache_data
def load_data():
    data = pd.read_csv("fake_or_real_news.csv")
    data['fake'] = data['label'].apply(lambda x: 0 if x == 'REAL' else 1)
    return data

# Module 3: Select Vectorizer and Classifier
def select_model():
    vectorizer_type = st.sidebar.selectbox("Select Vectorizer", ["TF-IDF", "Bag of Words"])
    classifier_type = st.sidebar.selectbox("Select Classifier", ["Linear SVM", "Naive Bayes"])

    vectorizer = None
    if vectorizer_type == "TF-IDF":
        vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
    elif vectorizer_type == "Bag of Words":
        vectorizer = CountVectorizer(stop_words='english', max_df=0.7)

    classifier = None
    if classifier_type == "Linear SVM":
        classifier = LinearSVC()
    elif classifier_type == "Naive Bayes":
        classifier = MultinomialNB()

    return vectorizer, classifier

# Module 4: Train the model
@st.cache(suppress_st_warning=True)
def train_model(data, vectorizer, classifier):
    x_vectorized = vectorizer.fit_transform(data['text'])
    clf = classifier.fit(x_vectorized, data['fake'])
    return vectorizer, clf

# Module 5: Streamlit app
def main():
    # Set page configuration
    page_icon = ":metro:"
    layout = "wide"
    page_title = "Fake News Detection"
    st.set_page_config(page_title=page_title, page_icon=page_icon, layout=layout)

    # Streamlit app
    st.title(page_title + " " + page_icon)
    st.lottie("https://lottie.host/bd0c4818-c5a6-4e42-b407-746bc448c2c7/ipVUdgFncO.json",
width=200, height=200)

    # --- HIDE STREAMLIT STYLE ---
    hide_st_style = """
    <style>
    #MainMenu {visibility: hidden;}
    footer {visibility: hidden;}
    header {visibility: hidden;}
    </style>
    """
    st.markdown(hide_st_style, unsafe_allow_html=True)

    # Load data
    data = load_data()

    # Select vectorizer and classifier
    vectorizer, classifier = select_model()

    # Text input for user to input news article
    user_input = st.text_area("Enter your news article here:")

    # When user submits the input
    if st.button("Check"):
        # Train the model and get the fitted vectorizer
        vectorizer, clf = train_model(data, vectorizer, classifier)
        # Vectorize the user input using the fitted vectorizer
        input_vectorized = vectorizer.transform([user_input])

        # Predict the label of the input
        prediction = clf.predict(input_vectorized)

        # Convert prediction to integer for interpretation
        result = int(prediction[0])

        # Display the result
        if result == 1:
            st.error("This news article is likely fake!")
        else:
            st.success("This news article seems to be real.")

    st.markdown("*Created with enthusiasm by Harshika & Aryan*")

# Run the Streamlit app
if __name__ == "__main__":
    main()
##run with command streamlit run main.py --client.showErrorDetails=false to remove cache
error message on streamlit interface
```