St. Thomas' College of Engineering and Technology

Department of Computer Science and Engineering

Fake News Detection Using Pattern Recognition

Prepared By

Student Name	University Registration No.
Madhurima Ranjit	016578 OF 2019-20
Aniket Sinha Roy	017301 OF 2019-20
Subham Tripathi	017343 OF 2019-20

Under the esteemed guidance of

Mrs. Ranjita Chowdhury

(Assistant Professor of Computer Science & Engineering Department)

Project Report

Submitted in the partial fulfillment of the requirement for the degree of

B. Tech. in Computer Science and Engineering

Affiliated to

Maulana Abul Kalam Azad University of Technology, West Bengal

2022-2023

St. Thomas' College of Engineering and Technology

Department of Computer Science and Engineering

This is to certify that the work in preparing the project entitled Fake News Detection Using Pattern Recognition has been carried out by **Madhurima Ranjit (University Roll No: 12200119030), Aniket Sinha Roy (University Roll No: 12200119019), and Subham Tripathi (University Roll No: 12200119018)** under my guidance during the session 2022-23 and accepted in the partial fulfilment of the requirement for the degree of Bachelor of Technology in Computer Science & Engineering.

Signature

Mrs. Ranjita Chowdhury
(Assistant Professor)
Department of Computer Science & Engineering
St. Thomas' College of Engineering
and Technology

Signature

Dr. Mousumi Dutt
(Head of The Department)
Department of Computer Science & Engineering
St. Thomas' College of Engineering
and Technology

St. Thomas’ College of Engineering and Technology

Department of Computer Science and Engineering

ACKNOWLEDGMENT

Inspiration and motivation have always played a key role in the success of any venture. During the work, we faced many challenges due to our lack of knowledge and experience but our Project Mentor, Mrs. Ranjita Chowdhury helped us get over all those difficulties and in the final compilation of our idea into a shaped sculpture.

We express our sincere gratitude to Mrs. Ranjita Chowdhury, who has guided us throughout the project and has given us her valuable suggestions for the successful completion of the project. She has helped us understand the intricate issues involved with the objective of the project which would have been overlooked otherwise.

We respect and thank Mrs. Ranjita Chowdhury for taking a keen interest in our project and for providing us with all the necessary resources and information for developing a good system in spite of having such a busy schedule.

We are also thankful to the entire Computer Science & Engineering Department for providing us with the technical support to carry out the project work, to let us use the Project Lab, and for guiding us at each & every step during the project work.

____
Madhurima Ranjit
University Roll No: 12200119030

____
Aniket Sinha Roy
University Roll No: 12200119019

____
Subham Tripathi
University Roll No: 12200119018

St. Thomas' College of Engineering and Technology

Department of Computer Science and Engineering

DECLARATION

We declare that this written submission represents our ideas in our own words and we have adequately cited and referenced the original sources. We also declare that we have adhered to all the principles of academic honesty and integrity and have not misrepresented, fabricated, or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Madhurima Ranjit
University Roll No: 12200119030

Aniket Sinha Roy
University Roll No: 12200119019

Subham Tripathi
University Roll No: 12200119018

Table Of Content

I. Preamble

I.I Vision and Mission

Vision of the Institute

To evolve as an industry-oriented, research-based Institution for creative solutions in various engineering domains, with an ultimate objective of meeting technological challenges faced by the Nation and the Society.

Mission of the Institute

- To enhance the quality of engineering education and delivery through accessible, comprehensive, and research-oriented teaching-learning-assessment processes in the state-of-art environment.
- To create opportunities for students and faculty members to acquire professional knowledge and develop managerial, entrepreneurial, and social attitudes with highly ethical and moral values.
- To satisfy the ever-changing needs of the nation with respect to evolution and absorption of sustainable and environment-friendly technologies for effective creation of knowledge-based society in the global era.

Vision of the Department

To continually improve upon the teaching-learning processes and research with a goal to develop quality technical manpower with sound academic and practical experience, who can respond to challenges and changes happening dynamically in Computer Science and Engineering.

Mission of the Department

- To inspire the students to work with the latest tools and to make them industry ready.
- To impart research-based technical knowledge.
- To groom the department as a learning centre to inculcate advanced technologies in Computer Science and Engineering with social and environmental awareness.

I.II Program Outcome and Program-Specific Outcome

Program Outcome (PO)

- **PO1:** Engineering Knowledge- Apply the knowledge of mathematics, science, engineering fundamentals and engineering specialization to the solution of complex engineering problems.
- **PO2:** Problem Analysis- Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering science.
- **PO3:** Design & Development of Solutions- Design solutions for complex engineering problems and design system components, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO4:** Conduct Investigations of Complex Problems- Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO5:** Modern Tool Usage- Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- **PO6:** The Engineer and Society- Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO7:** Environment and Sustainability- Understand the impact of professional engineering solutions in social and environmental context and demonstrate the knowledge of, and need for sustainable development.
- **PO8:** Ethics- Apply ethical principles and commit to professional ethics and responsibilities and norm of engineering practice.
- **PO9:** Individual and Team Work- Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings.
- **PO10:** Communication- Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instruction.

- **PO11:** Project Management and Finance- Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team to manage projects and in multidisciplinary environments.
- **PO12:** Life-long Learning- Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcome (PSO)

- **PSO1:** Programming skills- Apply fundamental knowledge and programming aptitude to identify, design and solve real-life problems.
- **PSO2:** Professional skills- Students shall understand, analyze and develop software solutions to meet the requirements of industry and society.
- **PSO3:** Competency- Students will be competent for competitive examinations for employment, higher studies and research.

I.III PO and PSO mapping with justification

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
Fake News Detection Using Pattern Recognition	2	3	3	3	-	3	-	3	2	3	2	3	3	3	2

Justification

- **PO 1:** The project uses the application of supervised learning algorithms hence statistics are being used. Hence the knowledge of mathematics and engineering has been applied.

- **PO 2:** A literature survey has been done to understand the performance of different supervised learning algorithms on the same dataset.

- **PO 3:** This project provides a solution to identify whether a news article is fake or not. An algorithm has been designed to solve the problem.

- **PO 4:** During the literature survey, we have gathered different research-based knowledge which has been implemented in this project.
- **PO 6:** This project will help society to find whether the news is fake or not as a result it will help society not to get influenced by fake news.
- **PO 8:** We shall follow professional ethics and not submit the work of other individuals in the project. We shall develop our own project by fair means and credit authors wherever required.
- **PO 9:** The project has required the team members to function effectively as individuals as well as communicate and coordinate among themselves to ensure a smooth flow of work.
- **PO 10:** Team discussion at different stages has been greatly beneficial to design this project. Good communication has been necessary to avoid any conflicts or confusion during the course of the project.
- **PO 11:** In this project, we have broken down the project into project management five stages which helped in the easy continuation of the problem.
- **PO 12:** Identifying fake news will always be needed by the society and this project has a scope to develop in the future.
- **PSO 1:** Programming knowledge has been required in this project to implement machine learning algorithms to obtain the desired result.
- **PSO 2:** The solution provided by the project will meet the requirements of better and more accurate prediction of fake news detection.
- **PSO 3:** Further research work can be done on this particular topic.

Chapter 1: Introduction

The advent of the World Wide Web and the rapid adoption of social media platforms paved the way for information dissemination that has never been witnessed in human history before. Besides other use cases, news outlets benefitted from the widespread use of social media platforms by providing updated news in near real-time to its subscribers. The news media evolved from newspapers, tabloids, and magazines to a digital form such as online news platforms, blogs, social media feeds, and other digital media formats. It became easier for consumers to acquire the latest news at their fingertips. Facebook referrals account for 70% of traffic to news websites. These social media platforms in their current state are extremely powerful and useful for their ability to allow users to discuss and share ideas and debate over issues such as democracy, education, politics and health. However, such platforms are also used with a negative perspective by certain entities creating biased opinions, manipulating mindsets, and spreading satire or absurdity. The phenomenon is commonly known as fake news. Fake News contains misleading information that could be checked. Here we are using different techniques and algorithms to detect if a news is fake or real.

1.1 Objective of the Project

The goal is to determine the authenticity of a news article using supervised learning techniques. This approach involves training a model with labeled examples of both real and fake news articles. The model then uses this knowledge to classify new articles as either genuine or false based on their characteristics. The aim is to create a reliable and efficient system for detecting fake news in today's era of misinformation.

1.2 Brief Description of Project

The main objective of this project is to identify whether a news article is real or fake. Fake news can cause misinformation and confusion on a particular topic, making it critical to determine the authenticity of news articles. To achieve this objective, the project begins by cleaning and processing the dataset to ensure error-free training and testing.

The next step involves training various supervised models using the pre-processed dataset. The project utilizes a range of models, including RNN, logistic regression, passive aggressive, and many more, to identify the best-performing model for the task. The aim is to select the model with the highest accuracy in detecting fake news.

Once the model has been selected, it can be used to determine the authenticity of new news articles or text articles. By analyzing the characteristics of the article, the model can classify it as either real or fake. This can help in preventing the spread of misinformation and the negative impact it can have on individuals and society.

In conclusion, identifying fake news is crucial in today's era of misinformation. This project utilizes supervised learning algorithms to train and select the best-performing model for the task of detecting fake news. By analyzing the characteristics of news articles, the model can classify them as real or fake, which can help in preventing the spread of misinformation and promoting accurate information.

1.3 Tools & Platform

Throughout the journey of completing this groundbreaking project on detecting fake news, we had the pleasure of utilizing a plethora of cutting-edge tools and platforms. Our toolkit included a wide range of innovative technologies, each of which played a crucial role in enabling us to successfully achieve our objectives. From state-of-the-art machine learning algorithms to advanced data cleaning and processing tools, our arsenal was comprehensive and diverse. We made use of several industry-leading software applications and programming languages to create a sophisticated and intelligent system for detecting fake news articles. With such a powerful toolkit at our disposal, we were able to create a highly accurate and reliable platform that will have a positive impact on combating the spread of misinformation. Here is a list of tools and platform that we used:

1.3.1 Python Programming Language

Python is a high-level, general-purpose programming language known for its simplicity, readability, and flexibility. It supports multiple programming paradigms, including object-oriented, imperative, and functional programming. Python's extensive standard library and wide range of third-party modules make it a popular choice for diverse applications, from data analysis to web development.

1.3.2 Google Colab

Google Colab is a cloud-based, Jupyter notebook environment that allows users to run Python code and perform data analysis and machine learning tasks. It provides free access to a powerful computing environment with pre-installed libraries and GPU support, making it popular among researchers, educators, and developers.

1.3.3 Jupyter Notebook

Jupyter Notebook is an open-source web-based interactive development environment (IDE) that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It supports a wide range of programming languages and is popular among data scientists and researchers.

1.3.4 Scikit-learn(sklearn) in Python

Scikit-learn (or sklearn) is a powerful open-source machine learning library for the Python programming language. It provides efficient and user-friendly implementations of a wide range of classification, regression, clustering, and dimensionality reduction algorithms, as well as tools for data preprocessing, model selection, and performance evaluation. With its extensive documentation, active community, and wide range of supported models, sklearn is a popular choice for building and deploying machine learning pipelines in Python.

1.3.5 NumPy in Python

NumPy is a powerful library for numerical computing in Python. It provides a high-performance array object, along with a collection of functions for operating on arrays and performing mathematical operations. NumPy's arrays are much faster and more memory-efficient than Python's built-in lists, making it an essential tool for scientific computing, data analysis, and machine learning.

1.3.6 Pandas in Python

Pandas is a powerful open-source data manipulation and analysis library for Python. It provides easy-to-use data structures and data analysis tools that enable users to quickly and efficiently clean, transform, and analyze data. Pandas is widely used in data science, finance, economics, and many other fields where data manipulation and analysis are essential.

1.3.7 NLTK (Natural Language Toolkit) in Python

The Natural Language Toolkit (NLTK) is a popular Python library designed to facilitate the processing and analysis of natural language data. It includes a wide range of modules and tools for tasks such as tokenization, part-of-speech tagging, stemming, and sentiment analysis. NLTK is widely used in research and education, as well as in industry applications such as chatbots and text classification systems.

1.3.8 Matplotlib in Python

Matplotlib is a popular data visualization library in Python that provides a wide range of tools for creating 2D and 3D plots, histograms, scatter plots, bar charts, and more. It offers a flexible and easy-to-use interface for customizing the appearance and style of visualizations and can be used in conjunction with other data analysis libraries, such as NumPy and Pandas, to create powerful and informative data visualizations.

1.3.9 Gensim in Python

Gensim is a popular open-source Python library for topic modeling and natural language processing. It allows users to easily build, train, and evaluate various topic models, such as Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP). Gensim also provides a range of utilities for text preprocessing, similarity calculations, and document indexing. It is widely used in academia and industry for analyzing large volumes of text data.

1.3.10 TensorFlow in Python

TensorFlow is a powerful open-source machine-learning framework for Python. It allows users to build and train various types of machine learning models, including neural networks, using high-level APIs. TensorFlow offers a range of tools for model development, optimization, and deployment, making it a popular choice for data scientists and machine learning engineers.

1.3.11 Pickle in Python

Pickle is a Python module used for serializing and de-serializing objects. It converts a Python object hierarchy into a byte stream that can be stored or transmitted and then re-constructs the object hierarchy from the byte stream. Pickle can be used to save program state, pass data between processes, and store persistent objects in a database.

1.3.12 HTML

HTML (Hypertext Markup Language) is a programming language used to create and structure content for the World Wide Web. It is the standard markup language for creating web pages and web applications. HTML is a markup language, meaning it uses tags and attributes to define the structure and content of a web page. HTML documents can be viewed in web browsers, which interpret the code and display the content as a web page. HTML is a fundamental component of web development and is often used in conjunction with other technologies like CSS and JavaScript.

1.3.13 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in markup language. CSS enables developers to control the appearance of web pages, including layout, colors, fonts, and animations. CSS separates the presentation from the content, making it easier to maintain and modify web pages. It is supported by all modern web browsers and is a fundamental tool for web design and development.

1.3.14 Flask Web Framework

Flask is a popular web framework for Python, designed to create simple, yet powerful web applications. It provides a lightweight, flexible, and easy-to-use framework that supports rapid development of web applications. With Flask, developers can build web applications that are scalable, maintainable, and secure, with features like dynamic templates, URL routing, and support for various database systems. Flask is widely used in the development of web applications and APIs, and is a great choice for both beginners and experienced developers.

1.3.15 GitHub

GitHub is a web-based platform that provides a version control system and collaboration tools for software development projects. It allows developers to host, manage, and track changes to their code repositories. With GitHub, developers can create and collaborate on projects, contribute to open-source projects, and track issues and bugs. It offers features like code review, pull requests, and branching, which facilitate collaborative development and ensure code quality. GitHub also enables continuous integration and deployment by integrating with popular tools like Jenkins and Travis CI. Moreover, GitHub provides a platform for community interaction, allowing developers to discover and contribute to a vast range of projects. It serves as a hub for sharing and accessing code, fostering collaboration and knowledge exchange within the software development community.

1.3.16 VSCode

Visual Studio Code, commonly referred to as VS Code, is a lightweight, open-source code editor developed by Microsoft. It has gained significant popularity among developers due to its powerful features and extensibility. VS Code supports a wide range of programming languages and offers a rich ecosystem of extensions, enabling developers to customize and enhance their coding experience. The editor provides features like syntax highlighting, intelligent code completion, debugging, and version control integration. It also supports key productivity features such as multiple cursors, code snippets, and built-in terminal access. VS Code's intuitive user interface and highly customizable layout make it suitable for various development workflows. Developers can install extensions to add functionality for specific languages, frameworks, or tools, tailoring their environment to their needs. Additionally, VS Code's integrated marketplace makes it easy to discover and install extensions created by the community. With its speed, versatility, and extensive ecosystem, Visual Studio Code has become a go-to choice for developers across different platforms and programming languages.

1.3.17 Kaggle

Kaggle is a popular online platform for data science and machine learning competitions, as well as a vast repository of datasets and notebooks. It provides a space for data scientists and machine learning practitioners to solve complex problems, compete, and share insights. Kaggle hosts competitions where participants work on real-world data problems, striving to build the best predictive models. These competitions attract a diverse community of data enthusiasts, fostering knowledge sharing and collaboration. Kaggle also offers a comprehensive set of tools and resources, including datasets, notebooks, and kernels. Users can access and explore a wide range of datasets from various domains, allowing them to practice and develop their data analysis and modeling skills. Kaggle kernels provide a Jupyter Notebook environment with built-in computational resources, enabling users to write, execute, and share code, visualizations, and explanations. The platform also hosts tutorials, forums, and discussions, creating a supportive community for learning and professional growth in the field of data science and machine learning.

1.4 Project Organization

The project report is organized as follows. Chapter 2 consists of the literature review which tells about the work which is done till now. After that chapter 3 contains the concept and problem analysis. Chapter 3 is mainly subdivided into two sub-chapter. The first one consists of problem analysis which briefly tells about the problem and the second one tells about the concepts that we used to solve the problem. After that chapter 4 contains design and methodology which tells how the project has been planned. Chapter 4 has sub-chapters named front-end and back-end which tells how the front-end and back-end of the web site was planned. It also has deployment and integration testing as sub-chapters which tells about how the model was deployed and how front-end and back-end are integrated. Chapter 5 contains the important coding portion done during front-end and back-end design. Chapter 6 contains the result and discussion related to the result by comparing accuracy rate of different models. At the end chapter 7 contains the conclusion and future work of the project.

1.5 Project Timeline

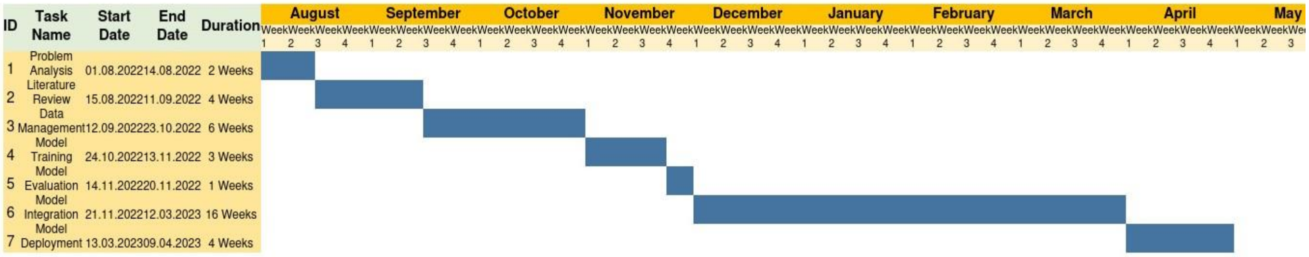
In project management, a project timeline is a crucial tool that provides a comprehensive view of the project plan in one place. It is a visual representation of tasks or activities arranged in chronological order, allowing project managers to track the project's progress from start to finish. A project timeline provides an in-depth overview of the entire project, including the duration, deadlines, and dependencies of each task.

For our project, we used a Gantt chart to illustrate the timeline of activities that have been completed and the future activities that need to be done. The Gantt chart is a popular tool for creating project timelines, and it is an effective way to visually display the project plan.

The timeline is divided into different phases, with all activities allocated to each phase corresponding to the required time. This approach allows the project manager to view the project's progress at a glance and identify any potential delays or roadblocks.

The Gantt chart used in our project includes a range of information, including task names, start and end dates, duration, and progress. It also includes information on the resources assigned to each task and any dependencies between tasks. This information enables project managers to effectively manage and allocate resources, identify potential bottlenecks, and ensure that the project stays on track.

Overall, the project timeline is an essential tool for project managers, enabling them to view the entirety of the project plan in one place. By using a Gantt chart, we were able to illustrate the project's timeline effectively, providing a clear view of the progress made so far and the activities that need to be completed in the future. This allows us to manage the project effectively and ensure that it is completed on time.



ID	Task Name	Start Date	End Date	Duration
1	Problem Analysis	01.08.2022	14.08.2022	2 Weeks
2	Literature Review	15.08.2022	11.09.2022	4 Weeks
3	Data Management	12.09.2022	23.10.2022	6 Weeks
4	Model Training	24.10.2022	13.11.2022	3 Weeks
5	Model Evaluation	14.11.2022	20.11.2022	1 Weeks
6	Model Integration	21.11.2022	12.03.2023	16 Weeks
7	Model Deployment	13.03.2023	09.04.2023	4 Weeks

Fig.1: Gantt Chart to monitor Activity and Planning

Chapter 2: Literature Survey

In[1] There are four types of models that have been applied here. Two different types of pre-processing are done on the dataset. In one pre-processing step count, a vectorizer is used. A simple way to both tokenize a collection of text documents and build a vocabulary of known words is provided by it. New documents using that vocabulary are encoded by it. In another pre-processing technique, TFIDF vectorizer is used that transforms text to feature vectors that can be used as input to estimator vocabulary that converts each token (word) to feature index in the matrix, each unique token gets a feature index. **In[7]** Two different types of training models are used to train these two different pre-processed datasets, one is Naive Bayes Classifier and another one is Passive-Aggressive Classifier. A Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. The crux of the classifier is based on the Bayes theorem. **In[10]** The passive-aggressive algorithms are a family of algorithms for large-scale learning. They are similar to the Perceptron in that they do not require a learning rate. The Naive Bayes classifier combined with the count vectorizer has given an accuracy level of 87%. The Naive Bayes classifier combined with the TFIDF vectorizer has given an accuracy level of 80%. **In[15]** The passive-aggressive classifier combined with the count vectorizer has given an accuracy level of 89%. The passive-aggressive classifier combined with the TFIDF vectorizer has given an accuracy level of 92%.

In[2] The process of making the model is done in six stages. At first, the data management phase is done where the dataset is collected and all the pre-procession is done. Term Frequency-Inverse Document Frequency (TF-IDF) is used to convert all the sentences of articles into a structured format. TFIDF works by proportionally increasing the number of times a word appears in the document but is counterbalanced by the number of documents in which it is present. Words that are very common are not given high priority. In the second stage, the model is trained with the help of logistic regression on the pre-processed data. Logistic regression estimates the probability of an event occurring, based on a given dataset of independent variables. The outcome is a probability, so the dependent variable is bounded between 0 and 1. **In[11]** Logistic regression uses a sigmoid function to convert the score into probability. The trained model is then saved using a .sav file. In the third stage, the model is evaluated by seeing the accuracy rate on test data and F1 score. In the fifth stage, **In[8]** the model has been integrated with the web portal to allow entry of news contents or news URL. The saved model is passed through API service which acts as a web service to the portal. In the last stage, the model is applied to new or fresh data and outcomes are monitored. The model gives a training accuracy of 99.2% and testing accuracy of 97% according to the report.

In[3] At first, two datasets, i.e. Fake and True are cleaned. Empty rows and the rows with some other errors are removed. The extra columns are also removed from the datasets. Then a class level column is added inside the datasets. After that these datasets are merged together. **In[12]** All the contents of the merged dataset are converted into lowercase characters. All special characters are also removed from the dataset. The 'title' and 'text' fields are merged together into a single column. After that the data is being prepared and transformed into a context which the machine can understand. For that the data are transformed into a list of vectors. Then the texts are tokenized by turning it into a list of sequences. Each word inside the data has its own sequence. The most frequent sequences are calculated. They are removed from the data. Then LSTM is used which remembers all the past knowledge that the network has seen so far. It also can forget irrelevant data. After that, the sequence vector of the news content is being trained using fed with sequence vectors of the news content. After training and fitting the model we got an accuracy of more than 98% for a dataset having 22000 news data. In order to test our data, the text news must be converted into sequences and after that, the model will predict the news.

In[4] According to the paper at first, all the pre-processing is done on the dataset. Tokenizing and stemming are the main pre-processing steps. Stemming is the process of reducing inflected words to their word stem, base, or root form—generally, a written word form and tokenization is the process of replacing sensitive data with unique identification symbols that retain all the essential information about the data without compromising its security. After all the pre-processing feature selection is done with the help of the TFIDF vectorizer or count vectorizer. **In[6]** the model is trained by logistic regression. It is a machine learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 or 0. The accuracy given by this model is 72% according to the paper.

In[5] According to the paper, the noises like ids, dots, commas, quotations, unwanted columns are removed from the dataset. Then using POS, the dataset is being turned into tokens and statistical values. Then unigram and bigram features are extracted using the TFIDF vectorizer function of python sklearn. Then the dataset is divided into two parts: 70% for training and 30% for testing. Then six classifier algorithms are used to produce a classification model. The algorithms are XGboost, Random Forests, Naive Bayes, K-Nearest-Neighbors, Decision Tree and SVM. Then the precision of those models is calculated using the test portion of the dataset and confusion matrix is being produced. Then the accuracy of those models is compared. The model with the highest accuracy is chosen. The highest accuracy can go up to 92%.

Dataset: In **In[1]** one dataset has been used having attributes ‘title’, ‘text’, ‘publishing date’, and ‘label’. The dataset has been taken from Kaggle. The size of the dataset is 25 MB having 20000 rows of data. In **In[2]** one dataset has been used having attributes ‘title’, ‘text’, ‘publishing date’, ‘author’, and ‘label’. The dataset has been taken from Kaggle. The size of the dataset is 96 MB having at least 25000 rows of data. In **In[3]** two datasets have been used. One for the fake news and another for the real news. Both datasets contain ‘title’, ‘text’, ‘subject’, and ‘date’ as attributes. The size of the real news dataset is 61 MB having at least 20000 rows of data and the size of the fake news dataset is 59 MB having at least 19000 rows of data. In **In[4]** three datasets are used. One dataset named as train.csv is used for training purposes which has attributes ‘title’, ‘text’, ‘author’, and ‘label’. The size of the dataset is 96 MB having 25000 rows of data. Another dataset named as test.csv is used for getting accuracy level of the model. The dataset has attributes named ‘title’, ‘text’, and ‘author’. The size of the dataset is 25 Mb having 5000 rows of data. Another dataset named submit.csv has the mapping of the labels to the dataset test.csv. The size of this file is less than 15 MB.

Chapter 3: Concepts and Problem Analysis

3.1 Problem Analysis

Fake news is a term used to describe news articles that are intentionally and verifiably false, with the aim of manipulating people's perceptions of real facts, events, and statements. The term has gained widespread usage in recent years as the proliferation of social media and online news sources has made it easier for misinformation to spread rapidly.

Fake news is different from satire or parody, which are often created for comedic or entertainment purposes. Rather, fake news is a deliberate attempt to deceive people by presenting information as news that is known by its promoter to be false based on facts that are demonstrably incorrect or events that did not happen. The spread of fake news is particularly concerning because it can lead people to make decisions based on false information, which can have serious consequences for individuals and society as a whole.

The identification of fake news is an important challenge for individuals and society as a whole. By understanding the characteristics of fake news and the ways in which it spreads, we can work to reduce its impact and promote a more informed and accurate public discourse. This requires a concerted effort by journalists, social media platforms, and individuals to verify information and promote fact-based reporting.

3.2 Concepts

The main concept that has been used in solving the problem of fake news detection is logistic regression, recurrent neural network, and passive-aggressive classifier. While applying these concepts we have also applied different Python modules. Here is a brief description of every concept that has been used in this project. Together, these approaches helped to develop a robust solution for identifying and combating fake news.

3.2.1 Logistic Regression

Logistic regression is a statistical model used to estimate the probability of an event occurring, based on a given dataset of independent variables. For example, it can be used to predict whether a person will vote or not based on their age, gender, income, and other factors. Since the outcome of logistic regression is a probability, the dependent variable is bounded between 0 and 1. To convert the value into a probability, logistic regression uses a special function called the sigmoid function. The sigmoid function takes any input value and maps it to a value between 0 and 1. This allows logistic regression to estimate the probability of an event occurring, given a set of input variables. Once the probabilities are calculated, the logistic regression model selects the class label with the higher probability as the predicted class. For example, if the probability of voting is higher than not voting, the model will predict that the person will vote. Logistic regression is a widely used and powerful tool in data analysis and machine learning, allowing us to make predictions based on complex data sets.

3.2.2 Passive Aggressive Classifier

Passive Aggressive Classifiers (PAC) are a type of online learning algorithm that is well-suited for systems that receive data in a continuous stream. In online learning, the model is trained and deployed in a way that allows it to continue learning as new data sets arrive. This makes PAC a good choice for applications such as real-time data analysis, where new data is continuously being generated and needs to be analyzed on the fly. PAC algorithms are typically used for large-scale learning tasks, where the training dataset is too large to fit into memory. Unlike other online learning algorithms, such as stochastic gradient descent, PAC algorithms do not require a learning rate. Instead, they use a "margin-based" approach to training, which means that the model is trained to maximize the margin between the decision boundary and the training data. In terms of their architecture, PAC algorithms are somewhat similar to the Perceptron model. However, unlike the Perceptron, PAC algorithms can handle non-linearly separable data, making them a more flexible choice for complex datasets. Overall, PAC algorithms are a powerful and versatile tool for online learning and are widely used in a variety of applications, including text classification and image recognition.

3.2.3 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural network that is specifically designed for processing sequential data. In traditional neural networks, all the inputs and outputs are independent of each other, and the network processes them in isolation. However, when dealing with sequential data, such as natural language text or time-series data, it's important to be able to take into account the context of previous inputs. RNNs solve this problem by introducing a hidden layer that retains information about previous inputs. In an RNN, the output from the previous time step is fed as input to the current time step. This allows the network to "remember" previous inputs and take them into account when making predictions about the current input. The most important feature of an RNN is the hidden state, which is a vector that captures information about the sequence processed so far. This hidden state is updated at each time step and serves as a kind of "memory" for the network. It allows the network to keep track of information that is relevant for predicting future inputs. However, traditional RNNs can suffer from the problem of vanishing gradients, which means that the gradient of the loss function with respect to the weights of the network can become very small and cause the network to stop learning. To address this problem, a variant of RNNs called Long Short-Term Memory (LSTM) was developed. LSTM networks are designed to retain information over longer periods of time and avoid the problem of vanishing gradients. They achieve this by introducing specialized memory cells that can selectively remember or forget information based on a set of learned parameters. This allows them to retain important information over long periods of time while discarding irrelevant information. LSTM networks have been used successfully in a wide range of applications, including natural language processing, speech recognition, and time-series prediction. In the context of fake news detection, an LSTM-based model can be trained on a dataset of news articles to identify patterns and features indicative of fake news. The model can then be used to classify new articles as either real or fake based on these learned patterns.

3.2.4 TFIDF Vectorization

TFIDF, or term frequency-inverse document frequency, is a technique used in natural language processing and information retrieval to determine the importance of a word in a document. The basic idea behind TFIDF is that the importance of a word increases proportionally with the number of times it appears in the document, but is offset by the frequency of the word in the corpus. In other words, TFIDF works by giving higher weight to words that appear frequently in a document, but less frequently across the corpus. This is because words that appear too frequently across all documents, such as 'this', 'are', and 'the', are unlikely to provide meaningful insights into the context of a particular document. The term frequency (TF) component of TFIDF measures the number of times a word appears in a document. It is computed by dividing the number of times a word appears in a document by the total number of words in the document. For example, if a word 'computer' appears 10 times in a document that contains 100 words, the TF of 'computer' in that document would be 0.1. The inverse document frequency (IDF) component of TFIDF measures the rarity of a word across the corpus. It is computed by taking the logarithm of the total number of documents divided by the number of documents that contain the word. The logarithm is added to dampen the importance of a very high value of IDF. For example, if there are 1 million documents in the corpus, and the word 'computer' appears in 1000 documents, the IDF of 'computer' would be log(1,000,000/1000) = 6. Finally, TFIDF is computed by multiplying the term frequency of a word with its inverse document frequency. This gives a higher weight to words that appear frequently in a document, but less frequently across the corpus. The resulting TFIDF score is a measure of the importance of a word in a document. TFIDF is commonly used in information retrieval, text mining, and natural language processing applications such as document classification, clustering, and search engines. It is a simple yet powerful technique that can help improve the accuracy and effectiveness of text-based algorithms.

3.3 Proposed Algorithms

3.3.1 Proposed Algorithm For Fake News Detection Using RNN Model

Step 1: At first step two datasets i.e. ‘Fake’ and ‘Real’ are read.

Step 2: Then they are divided according to their publisher names. Rows with no publishers are set as “unknown.”.

Step 3: Then the ‘title’ and ‘text’ column of each dataset is merged together into a single ‘text’ column. And a ‘class’ column is added(1 for real and 0 for fake).

Step 4: The ‘text’ and ‘class’ columns of two datasets are merged together into a single one.

Step 5: The characters of the ‘text’ part are converted into lowercase and all the special characters are removed.

Step 6: Each word inside the ‘text’ column is turned into a list and the words are converted into vectors.

Step 7: The words are then turned into the form of sequences.

Step 8: Then the most frequent words are padded into default value 0.

Step 9: Weight matrix is created using the vectors.

Step 10: A model is then created using the sequential function and the weight matrix is added.

Step 11: LSTM and Dense layers are added into the model.

Step 12: 70% of the data is used for training the model and the remaining 30% is used for testing.

Step 13: Then the accuracy level is calculated.

Step 14: Then the model is saved in hdf5 format. So that we don’t have to train the same model over and over again.

Step 15: In order to predict the results, first the saved model is compiled. And then text data is taken as input. The text data is converted into sequences and padded. Then the result is predicted.

3.3.2 Proposed Algorithm For Fake News Detection using Logistic Regression Model

Step 1: The training dataset is read having columns author, title, text, and labels.

Step 2: All the missing data are replaced with "empty string".

Step 3: A new column is created combining the author and title of the news and naming the column as authorAndTitle.

Step 4: Removing all kinds of punctuation from the data in column authorAndTitle and also changing all the alphabets into lowercase.

Step 5: Removing all kinds of stopwords and applying stemming to the column authorAndTitle.

Step 6: The data is then converted into numerical form and then converted to feature form.

Step 7: 70% of the data from column authorAndTitle is selected randomly which is used to train the logistic regression model.

Step 8: Rest of the 30% of the data is taken for testing purposes and the accuracy of the model is calculated.

Step 9: The model is then saved with the help of pickle module in python.

Step 10: New data is taken and the save model classify the data as either fake or real.

Chapter 4: Design & Methodology

The implementation can be divided into two parts:

a) **Front-End:** Designing user-interactive web pages using HTML and CSS.

b) **Back-End:** Designing the Logistic Regression model using python and connecting it with the front-end using Flask Web Framework

4.1 Front-End

The front-end consists of a web page which will take the input from users and display if the provided news is fake or legitimate.

The key points that are taken into consideration while designing the front-end are:

- ### User Interface (UI) Design:
  - ➢ Keeping the interface clean and intuitive, with a user-friendly layout.
  - ➢ Using appropriate colors and typography to enhance readability and visual appeal.
  - ➢ Clearly distinguishing between real and fake news results to avoid confusion.
- ### Input and Output Elements:
  - ➢ Providing a text input field for users to enter the news article they want to analyze.
  - ➢ Displaying the analysis results prominently, indicating whether the news is classified as real or fake.
- ### Error Handling and Validation:
  - ➢ Implementing input validation to ensure users provide valid inputs, such as proper text content.
  - ➢ Displaying clear error messages if invalid input is detected or if there are issues with the analysis process.
- ### Clear Instructions:
  - ➢ Including concise instructions on how to use the system effectively.
  - ➢ Offering contextual help to assist users in understanding the system's features and limitations.

- **Accessibility Considerations:**
  - Implementing accessibility features such as keyboard navigation, alt text for images, and adequate color contrast.
  - Ensuring that the frontend is compatible with screen readers.

Following are the three web pages designed to provide the users with interactive and easy to use platform:

4.1.1 Home Page

The Homepage is basically the first page that appears on the screen where the user will provide inputs. The components of the Home Page are as follows:

4.1.1.1 Title of the Website:

The home page contains a title "Fake News Detector" for recognition.

4.1.1.2 Heading:

The Heading is as same as the title as it gives the user a brief about the purpose of the web page.

4.1.1.3 Description:

A one-line description is provided for better understanding to the users.

4.1.1.4 Title:

This is a text type input box where the user is expected to provide the title of the news they want to check the legitimacy of. Clear instruction is written on the homepage.

4.1.1.5 Author:

This is a text type input box as well where the user is expected to write the source of the news, such as the author's name or the name of the newspaper from where the news is gathered or the name of the website where the news is encountered. Clear instruction is written on the homepage.

4.1.1.6 Detailed News:

This is a text type input box where the user is expected to provide the entire news or an excerpt of it. Clear instruction is given on the homepage. The size of the textarea can be adjusted according to the length of the news. Scroll button will be displayed if the length of the news exceeds the size of the textarea.

4.1.1.7 Verify:

This is a button on clicking which the data provided by the user will be fed to the Logistic Regression Machine Learning Model and after checking whether the news is fake or real, the corresponding result will be displayed on the screen. The button will pop upon clicking for visual confirmation of submission of the data provided.

4.1.2 Result Pages

Two result pages are designed using HTML and CSS. Their function and components are as follows:

4.1.2.1 Fake News:

When the news is detected as fake, the user will be redirected to this web page to display that the provided news is fake.

4.1.2.1 Real News:

When the news is detected as real, the user will be redirected to this web page to display that the provided news is real.

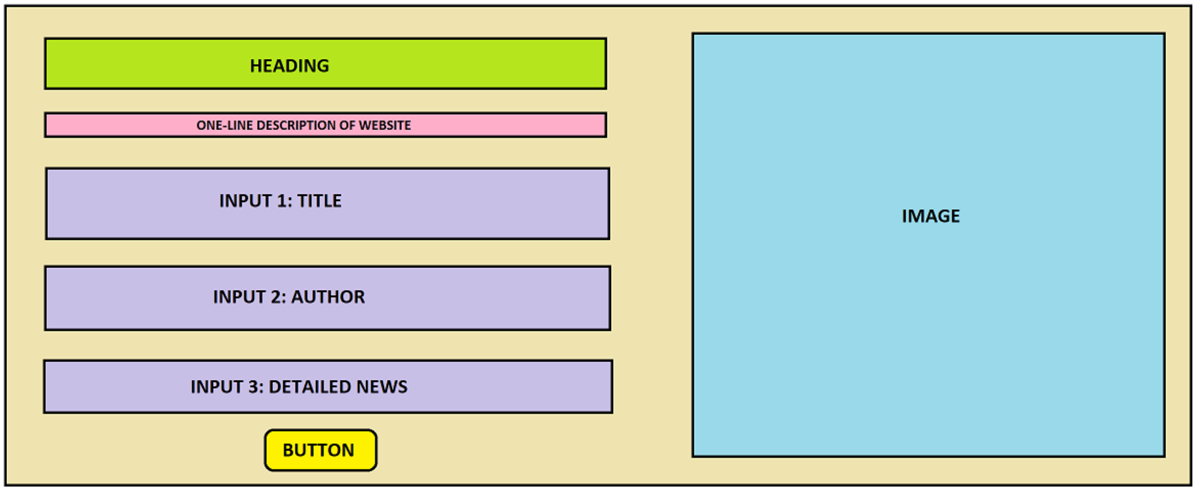


Fig 2. Wireframe Diagram of the Home Page

4.1.3 Styling

HTML only helps to build the skeleton of a web page, unless it is styled, the web page does not provide user-friendliness. Hence, the web page is styled using CSS. A light color is chosen as the background color. To make the input boxes prominent, dark colors and easily readable fonts are chosen which in turn ensures adequate color contrast. With the intention of designing the web pages more appealingly or visually eye-catching, images and borders are used.

4.2 Back-End

The back-end consists of the Machine Learning Model used to detect whether the news is fake or legitimate and the Flask Web Framework that connects the front-end with the Machine Learning Model and provides the user accurate outcome.

The basic steps of implementing the back-end are as follows:

4.2.1 Dataset Collection

Obtaining a comprehensive and representative dataset comprising a significant number of fake and legitimate news articles is the initial step in this process. We have considered using reputable fact-checking sources, known fake news websites, and reliable news sources for collecting the data. The selected dataset covers various topics and contexts.

To build an effective fake news detection system, you would typically need the following types of datasets:

4.2.1.1 Fake News Articles:

A dataset containing a significant number of labeled fake news articles is essential. These articles are sourced from known unreliable or deceptive sources, fact-checking organizations, or datasets specifically curated for fake news research. It is crucial to ensure that the labels indicating the articles' authenticity are accurate and reliable.

4.2.1.2 Legitimate News Articles:

A dataset comprising a substantial number of legitimate news articles from reputable and trusted sources is necessary. These articles cover a wide range of topics and should be verified as genuine and factually accurate. Including a diverse set of legitimate articles helps create a balanced dataset for training and evaluation.

4.2.1.3 Mixed News Articles:

In addition to labeled fake and legitimate news articles, it can be beneficial to include a set of news articles with ambiguous authenticity. These articles have undergone fact-checking processes, with conflicting assessments from multiple sources. Including such articles helps the model learn to handle uncertain cases and develop more nuanced decision-making capabilities.

4.2.1.4 Metadata and Contextual Information:

Alongside the news articles themselves, including metadata and contextual information can provide valuable insights. Therefore, information such as the source of the article, author details are included. Incorporating this information allows the model to consider the credibility of the sources and the temporal context in which the articles were published.

It is important to ensure that the collected datasets are representative, diverse, and balanced. Careful attention is given to the quality and accuracy of the labeling process to ensure reliable ground truth for training and evaluation.

4.2.2 Feature Extraction

Feature extraction plays a crucial role in fake news detection as it helps capture meaningful patterns and characteristics that can differentiate between fake and legitimate news articles. Here are some common types of features used in fake news detection:

4.2.2.1 Linguistic Features:

Linguistic features focus on the textual content of the news articles. They include:

- **Word Frequency:** Counting the occurrence of words in the article can provide insights. Certain words or phrases may be more prevalent in fake news articles, such as exaggerated claims, articles (a, an, the) etc.
- **N-grams:** Analyzing sequences of adjacent words (e.g., unigrams, bigrams, trigrams) can capture contextual information and language patterns.
- **Part-of-Speech (POS) Tags:** Identifying the grammatical category of each word (e.g., noun, verb, adjective) can help understand the syntactic structure and linguistic patterns.

- **Sentiment Analysis:** Analyzing the sentiment expressed in the article, such as positive or negative sentiment, can provide an additional perspective on the article's tone and credibility.

4.2.2.2 Stylistic Features:

Stylistic features focus on the writing style and characteristics of the article. They include:

- **Readability Scores:** Assessing the complexity of the article using readability measures like Flesch-Kincaid Grade Level or Coleman-Liau Index. Fake news articles may exhibit different readability patterns compared to legitimate ones.
- **Capitalization and Punctuation:** Examining the use of excessive capitalization, exclamation marks, or misleading punctuation can indicate sensationalism or clickbait.
- **Sentence Length and Structure:** Analyzing the length and structure of sentences, such as the average sentence length or the presence of fragmented sentences, can reveal stylistic differences.

4.2.2.3 Contextual Features:

Contextual features consider external factors and metadata associated with the articles. They include:

- **Source Reputation:** Assessing the credibility and reliability of the news source based on its reputation or bias. Fake news articles often originate from dubious or unreliable sources.
- **Author Details:** Investigating the credibility, expertise, and history of the article's author can provide insights into the authenticity of the content.

4.2.3 Data Preprocessing

Data preprocessing is an essential step in fake news detection to clean and prepare the data for analysis. It involves transforming raw data into a format that is suitable for feature extraction and model training. Here are some common data preprocessing techniques used in fake news detection:

4.2.3.1 Text Cleaning:

Irrelevant information from the text, such as HTML tags, URLs, special characters, or punctuation marks are removed. This ensures that the data is clean and consistent for further processing.

4.2.3.2 Tokenization:

The text is divided into individual words or tokens. This step breaks down the text into meaningful units, making it easier to analyze and extract features.

4.2.3.3 Stop Word Removal:

Removal of common words like "the," "is," or "and" that do not carry significant meaning in the analysis increases the accuracy and runtime of the model. Stop word removal reduces noise and focuses on more informative terms.

4.2.3.4 Lowercasing:

All text is converted to lowercase to ensure consistency in word representation. This prevents the model from treating the same word with different cases as different features.

4.2.3.5 Stemming or Lemmatization:

Reducing words to their root form to handle variations and improve feature consistency is undertaken. Stemming reduces words to their base form by removing prefixes or suffixes, while lemmatization maps words to their dictionary form.

4.2.3.6 Handling Imbalanced Classes:

If the dataset has imbalanced classes, where the number of fake and legitimate news articles is significantly different, undersampling technique is applied to balance the dataset and prevent bias in the model's training.

4.2.3.7 Vectorization:

The preprocessed texts are converted into numerical representations that can be used as features for machine learning algorithms. TF-IDF (Term Frequency-Inverse Document Frequency) approach is used for this method.

4.2.3.8 Data Splitting:

The preprocessed dataset is divided into training, validation, and testing sets. The training set is used to train the model, the validation set helps tune hyperparameters and assess model performance, and the testing set evaluates the final model's performance on unseen data.

These preprocessing steps help ensure the data is clean, standardized, and ready for feature extraction and subsequent model training and evaluation.

4.2.4 Machine Learning Model Selection

Select an appropriate machine learning algorithm for building the fake news classifier. Consider supervised learning algorithms such as logistic regression, support vector machines, decision trees, or ensemble methods like random forests or gradient boosting. Additionally, explore deep learning models like recurrent neural networks (RNNs) or transformers that can capture complex relationships within the data.

Logistic Regression Model:

Logistic regression is a popular statistical model used for binary classification tasks. It is particularly useful when the dependent variable is categorical and takes one of two possible outcomes.

Logistic regression makes several key assumptions:

- **Binary Outcome:** The dependent variable must be binary or dichotomous, meaning it can take only two distinct values, typically encoded as 0 and 1.
- **Linearity:** The relationship between the independent variables and the log-odds of the outcome should be linear. This assumption can be assessed by examining scatterplots or through techniques like residual analysis.
- **Independence of Observations:** Each observation in the dataset should be independent of others. This assumption ensures that the model estimates are not biased or influenced by correlations between observations.

- **No Multicollinearity:** Independent variables should not exhibit high levels of multicollinearity, which refers to strong correlations between predictors. Multicollinearity can lead to unstable estimates and challenges in interpreting the model.

Logistic regression models the relationship between the independent variables and the log-odds of the dependent variable using the logistic function, also known as the sigmoid function. The logistic function takes any real-valued number as input and maps it to a value between 0 and 1, representing the probability of the event occurring.

The logistic regression equation is expressed as:

$\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$

where logit(p) represents the log-odds of the dependent variable, p represents the probability of the event occurring, β_0 is the intercept, β_1, β_2, ..., β_p are the coefficients corresponding to the independent variables x_1, x_2, ..., x_p.

To convert the log-odds into probabilities, the logistic function is applied:

$p = 1 / (1 + \exp(-\text{logit}(p)))$

In logistic regression, the coefficients (β_0, β_1, β_2, ..., β_p) indicate the impact of the independent variables on the log-odds of the outcome. They can be interpreted as the change in the log-odds of the dependent variable for a one-unit change in the corresponding independent variable, while holding other variables constant.

The odds ratio (OR) is commonly used to interpret the coefficients. It represents the ratio of the odds of the outcome occurring for a one-unit increase in the independent variable compared to a one-unit decrease, all else being equal. An OR greater than 1 indicates a positive relationship, while an OR less than 1 indicates a negative relationship.

4.2.5 Training and Validation

The preprocessed dataset is split into training and validation sets. Using the training set the model is trained on the extracted features. Experimentation with different feature combinations to optimize the model's performance has occurred. With the validation set, the trained model is tested to assess its accuracy, precision, recall, and F1-score.

Two validation approaches are undertaken. They are as follows:

4.2.5.1 Cross-validation:

It is a really important step to ensure the model’s stability and generalizability. It mitigates overfitting and provides a more robust estimation of the model's performance by evaluating it on multiple subsets of the training data.

4.2.5.2 Iterative Refinement:

Iterating the training and validation process with new datasets, adjusting the feature extraction techniques, model selection, and hyperparameters based on the evaluation results helps to increase the model’s accuracy level. This iterative process improves the model's performance and ensures it is effectively capturing the characteristics of fake and legitimate news articles.

4.2.6 Evaluation

Once the fake news detection model has been validated, the next step is to evaluate its performance. Here are the evaluation steps:

4.2.6.1 Evaluation Metrics Calculation:

Various evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve are calculated. These metrics provide insights into the model's ability to correctly classify fake and legitimate news articles.

4.2.6.2 Confusion Matrix Analysis:

The confusion matrix is analyzed to understand the model's true positive, true negative, false positive, and false negative predictions. The confusion matrix provides a visual representation of the model's performance and can help identify specific types of misclassifications.

4.2.6.3 ROC Curve Analysis:

The Receiver Operating Characteristic (ROC) curve is analyzed to evaluate the model's performance over a range of classification thresholds. The ROC curve plots the true positive rate against the false positive rate at various threshold settings. The area under the ROC curve (AUC) provides a single metric to compare different models' performance.

4.2.6.4 Statistical Significance Testing:

Statistical significance testing is conducted to determine if the observed differences in the evaluation metrics between different models or methods are significant. This analysis can help select the best-performing model or feature extraction method.

4.2.6.5 Error Analysis:

Error analysis is performed to identify specific types of misclassifications and their underlying causes. This analysis can help identify areas for improvement in the model, such as missing features or inadequate training data.

4.2.6.6 Interpretability Analysis:

The model's interpretability is analyzed to understand how it makes predictions. This analysis can help identify the features that the model relies on to make its predictions and provide insights into the characteristics of fake and legitimate news articles.

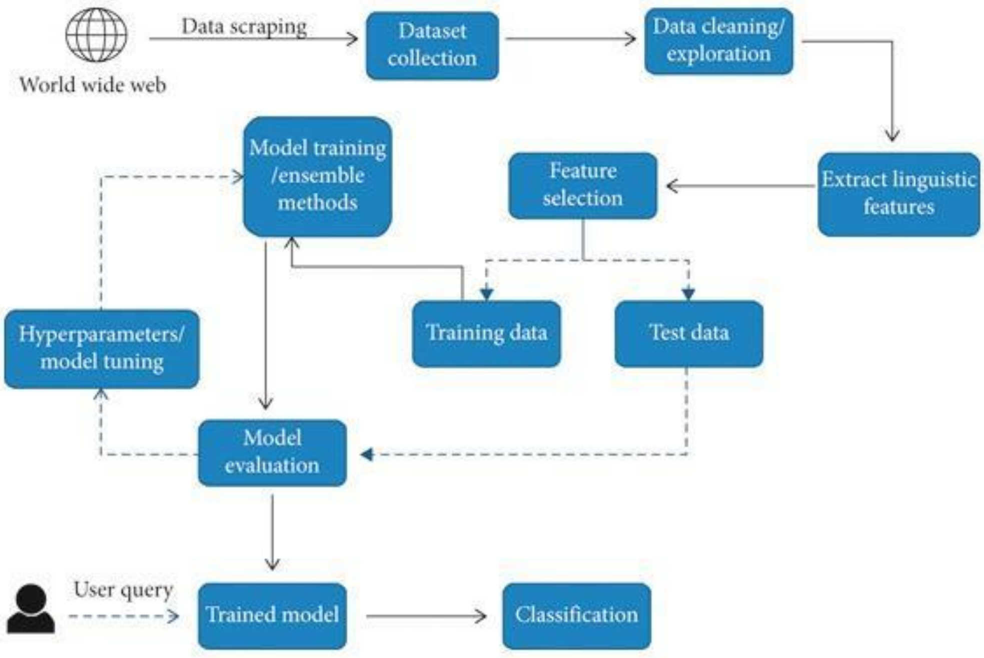


Fig. 3: Flow chart on the working principle of the Fake News Detection System

4.3 Deployment

Once satisfied with the model's performance, it is deployed into a production environment. A user-friendly interface is developed to allow users to input news articles and receive real-time predictions on their authenticity. In other words, the back-end, which is the fake news detection machine learning model and the front end, which is the web pages designed using HTML and CSS are developed separately. To connect these two units together to build the entire system, Flask Web Framework is used which is a python-based framework for deployment.

The data from the website is fetched using the POST method and they are stored in a NumPy array, each word as an array element. These data are fed to the machine learning model which is saved in pickle format. The model calculates the legitimacy of the news and based on the accuracy score, it returns 1 if the news is fake and 0 if legitimate.

If the value is 1, as known as fake news, the Fake News HTML page will be displayed using render template and if it is 0 that indicates legitimate news, the Real News HTML page will be displayed.

Chapter 5: Sample Codes

5.1 Logistic Regression Model

5.1.1 Data Cleaning

```
# counting the number of missing values in the dataset
df.isnull().sum()

# replacing the null values with empty string
df = df.fillna('')
```

5.1.2 Data Processing

```
import nltk
nltk.download('stopwords')
#downloading stopwords


port_stem = PorterStemmer()                        #Loading
porterStemmer() function to this variable port_stem


def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)        #removing
numbers and punctuations from content & replace it with empty string
    stemmed_content = stemmed_content.lower()
#converting everything to lowercase
    stemmed_content = stemmed_content.split()                #creating
a list ["the", "is",....] like this
    stemmed_content = [port_stem.stem(word) for word in
stemmed_content if not word in stopwords.words('english')] #removing
stopwords then performing stemming
    stemmed_content = ' '.join(stemmed_content)              #joining
the list ex. the is
    return stemmed_content

df['title'] = df['title'].apply(stemming)
df['author'] = df['author'].apply(stemming)
df['text'] = df['title'] + ' ' + df['author']
```

5.1.3 Model Training

```
x_train, x_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(x_train, y_train)

# evaluate the model on the testing data
y_pred = model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

5.1.4 Model Saving and Result Prediction

```
# save the model using pickle
filename = 'logisticRegressionSavedModelTitleAuthor.pkl'
with open(filename, 'wb') as file:
    pickle.dump(model, file)


# load the model and make predictions on new data
with open(filename, 'rb') as file:
    loaded_model = pickle.load(file)
new_text = 'Specter of Trump Loosens Tongues, if Not Purse Strings, in
Silicon Valley - The New York Times'
new_text = stemming(new_text)
new_text = vectorizer.transform([new_text])
prediction = loaded_model.predict(new_text)[0]
```

5.2 Recurrent Neural Network

5.2.1 Data Cleaning

```
unknown_publishers = []
for index, row in enumerate(real.text.values):
  try:
    record=row.split('-', maxsplit=1)
    record[1]
    assert(len(record[0])<120)
  except:
    unknown_publishers.append(index)
```

```
publisher=[]
tmp_text=[]


for index, row in enumerate(real.text.values):
  if index in unknown_publishers:
    tmp_text.append(row)
    publisher.append('Unknown')
  else:
    record = row.split('-', maxsplit=1)
    publisher.append(record[0].strip())
    tmp_text.append(record[1].strip())

real['text']=real['title']+ " "+real['text']
fake['text']=fake['title']+ " "+fake['text']
real['text']=real['text'].apply(lambda x: str(x).lower())
fake['text']=fake['text'].apply(lambda x: str(x).lower())
```

5.2.2 Data Preprocessing

```
real['class']=1
fake['class']=0
real=real[['text','class']]
fake=fake[['text','class']]
data=real.append(fake, ignore_index=True)
import preprocess_kgptalkie as ps
data['text']=data['text'].apply(lambda x: ps.remove_special_chars(x))
```

5.2.3 Data Vectorization

```
import gensim
y=data['class'].values
y=data['class'].values
X=[d.split() for d in data['text'].tolist()]
DIM = 100
w2v_model= gensim.models.Word2Vec(sentences= X, size= DIM, window= 10,
min_count=1)
tokenizer= Tokenizer()
tokenizer.fit_on_texts(X)
X = tokenizer.texts_to_sequences(X)
maxlen=1000
```

```
X=pad_sequences(X,maxlen=maxlen)
vocab_size = len(tokenizer.word_index) + 1
vocab = tokenizer.word_index
def get_weight_matrix(model):
  weight_matrix = np.zeros((vocab_size, DIM))
  for word, i in vocab.items():
    weight_matrix[i]=model.wv[word]
  return weight_matrix
embedding_vectors = get_weight_matrix(w2v_model)
```

5.2.4 Model Training and Saving

```
model = Sequential()
model.add(Embedding(vocab_size, output_dim = DIM, weights =
[embedding_vectors], input_length=maxlen, trainable= False))
model.add(LSTM(units=128))
model.add(Dense(1, activation = 'sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['acc'])
X_train, X_test, y_train, y_test = train_test_split(X,y)
model.fit(X_train, y_train, validation_split=0.3, epochs=6)
y_pred= (model.predict(X_test) >=0.5).astype(int)
from tensorflow.keras.models import load_model
model.save('/content/drive/MyDrive/SavedModel/tmsSavedModel.hdf5')
```

5.2.5 Model Loading and Testing

```
import tensorflow as tf
import gensim
from tensorflow.keras.preprocessing.text import Tokenizer #used to
tokenize the text data
from tensorflow.keras.preprocessing.sequence import pad_sequences
maxlen=1000
tokenizer= Tokenizer()
model_new=tf.keras.models.load_model("/content/drive/MyDrive/SavedMode
l/tmsSavedModel.hdf5")

model_new.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])

x=['Household incomes posted strong growth last year in more than a
dozen U.S. congressional districts where Republicans face stiff
challenges in November elections, according to a Reuters analysis of
Census Bureau data published on Thursday.']
x=['Pakistan is beset by violence in July after the military storms an
Islamabad mosque taken over by militant Islamists. On October 18, with
pressure rising for new elections, former premier Benazir Bhutto
returns from self-exile. A suicide bombing kills 139 people during her
homecoming parade. On November 3, President Pervez Musharraf imposes a
state of emergency, and then sets elections for January. The United
States, which has built its "war on terror" on an alliance with
Pakistan.']


tokenizer.fit_on_texts(x)
x=tokenizer.texts_to_sequences(x)
x=pad_sequences(x, maxlen=maxlen)
print(model_new.predict(x))
(model_new.predict(x)>=0.4).astype(int)
```

5.3 Front-End

5.3.1 HTML Code

5.3.1.1 Home Page

```
<body>
    <div id="det">
        <h1>Fake News Detector</h1>
        <img src="hr.png">
        <h4>A trusted website to determine whether a news is Real or
Fake.</h4>
    </div> <br>

    <form action="/predict" method="POST">
        <div class="t">
            <label for="title">Title:</label><br>
            <input type="text" id="title" name="title"
placeholder="Type the title of the news here...">
        </div>

        <div class="a">
            <label for="author">Author:</label><br>
            <input type="text" id="author" name="author"
placeholder="Type the author's name here...">
        </div>

        <div class="d">
            <label for="details">Detailed News:</label><br><br>
            <textarea id="details" name="details" placeholder="Type
the detailed news here..."></textarea><br>
        </div>

        <div class="b">
        <input type="submit" class="button" value="VERIFY">
    </div>
    </form>

</body>
```

5.3.1.2 Real News

```
h2{
   padding-top: 100px;
   text-align: center;
   color: green;
   }
<body>
    <h1>Fake News Detector</h1>
    <hr>
    <h2>REAL NEWS</h2>
</body>
```

5.3.1.3 Fake News

```
h2{
   padding-top: 100px;
   text-align: center;
   color: red;
   }
<body>
    <h1>Fake News Detector</h1>
    <hr>
    <h2>FAKE NEWS!!!</h2>
</body>
```

5.3.2 CSS Code: Home Page

```
body {
    background-color: rgba(254,250,239,255);
    background-image: url(bird.jpg);
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 680px 930px;
    background-position: right center;
}

#det {
    font-family: Arial, Helvetica, sans-serif;
    color: rgba(173, 65, 6, 0.974);
    padding-top: 20px;
    padding-right: 30px;
    padding-bottom: 10px;
    padding-left: 50px;
}

.t {
    font-size:large;
    position: relative;
    max-width: 213px;
    padding-right: 100px;
    padding-left: 50px;
    padding-bottom: 20px;
}

.a {
    font-size:large;
    position: relative;
    max-width: 213px;
    padding-left: 50px;
}

.d {
    font-size:large;
    position: relative;
    max-width: 213px;
    padding-left: 50px;
    padding-top: 20px;
}
```

```
.button{
    position: relative;
    padding-top: 50px;
    padding-left: 450px;
}

img{
    width: 950px;
    height: 2px
}

input[type=text], select {
    font-family: 'Courier New', Courier, monospace;
    width: 400%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    box-sizing: border-box;
}

textarea {
    width: 400%;
    height: 200px;
    padding: 12px 20px;
    box-sizing: border-box;
}

input[type=submit] {
    background-color: #aa4404;
    border: none;
    color: white;
    padding: 12px 20px;
    text-decoration: none;
    margin: 4px 2px;
    border-radius: 12px;
    cursor: pointer;
    box-shadow: 0 5px rgb(169, 162, 148);
  }
.button:hover {
    background-color: burlywood;
    color: #aa4404;
    font-weight: bold;
  }

  .button:active {
    background-color: burlywood;
    box-shadow: 0 5px #666;
    transform: translateY(4px);
  }

.b{
    display: flex;
    justify-content: center;
    align-items: center;
    height: 50px;
    width: 900px;
  }
```

5.4 Flask Web Framework

```
app=Flask(__name__)
@app.route('/')
def index_view():
    return render_template("index.html")

@app.route('/predict', methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        with open("logisticRegressionSavedModelTitleAuthor.pkl", "rb")
as file:
            loaded_model=pickle.load(file)
        vectorizer = TfidfVectorizer()
        t=request.form.get('title')
        a=request.form.get('author')
        text=t+' '+a
        new_text = stemming(text)
        with open('vectorizer.pkl', 'rb') as file:
            vectorizer = pickle.load(file)
        new_data_vectors = vectorizer.transform([new_text])
        predict = loaded_model.predict(new_data_vectors)
        if int(predict)== 0:
            return render_template("real.html", prediction=predict)
        else:
            return render_template("fake.html", prediction=predict)
if __name__=="__main__":
    app.run()
```

Chapter 6: Testing, Results, Discussion on Results

6.1 Testing

For the confirmation of the project's ability to sustain real life action, various types of tests and experiments have been designed and performed which are recorded to claim that the objective of the project has been achieved. They are as follows:

6.1.1 Unit Testing:

Testing individual components of the system, such as functions or methods in the machine learning model, to ensure they work correctly. The various methods or units are Data Cleaning, Feature Extraction, Data Preprocessing, training the model etc. Mocking or stubbing dependencies and input data to isolate the unit under test is an essential property to ensure while performing unit testing.

6.1.2 Functional Testing:

The functionality of the fake news detection system as a whole is validated. Various scenarios, such as entering different types of news articles, text inputs are handled and then cross-validated with the expected and obtained result to measure the accuracy of the system.

6.1.3 User Interface (UI) Testing:

Verification of the frontend web page for proper rendering, layout, readability and accessibility is performed. Correct functioning of user interactions, such as submitting inputs, receiving and displaying results, and handling errors are tested to provide user satisfaction.

6.1.4 Integration Testing:

The interaction between different components of the system, such as the integration between the Flask web framework and the machine learning model is tested. The data flow from the frontend to the backend and vice versa is tested and correct functioning of the communication and data processing is ensured.

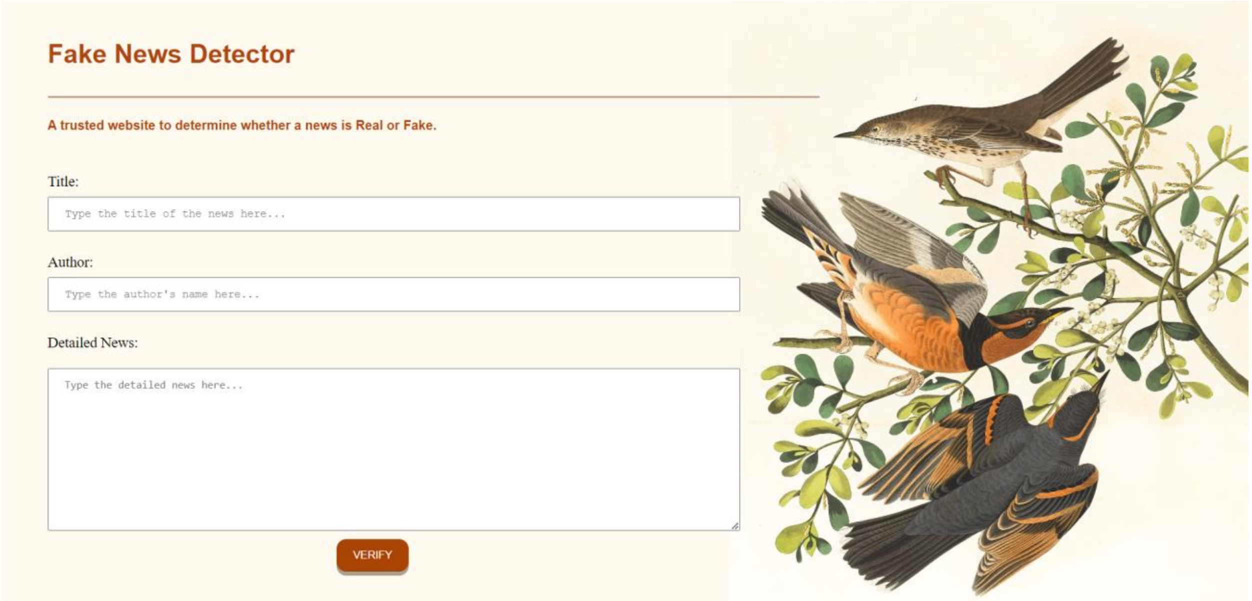
6.1.5 Performance Testing:

The system's performance under different workloads and stress conditions are assessed along with measurement of the response time of the web application, especially while handling large amounts of data or simultaneous requests. Potential bottlenecks or performance issues are identified and the system is optimized accordingly.

6.1.6 Compatibility Testing:

The system is tested on different web browsers such as Google Chrome, Microsoft Edge, Mozilla Firefox etc. to ensure cross-browser compatibility. Compatibility of the system is validated across different versions of Windows as well.

6.2 Results



Fake News Detector

A trusted website to determine whether a news is Real or Fake.

Title:

Type the title of the news here...

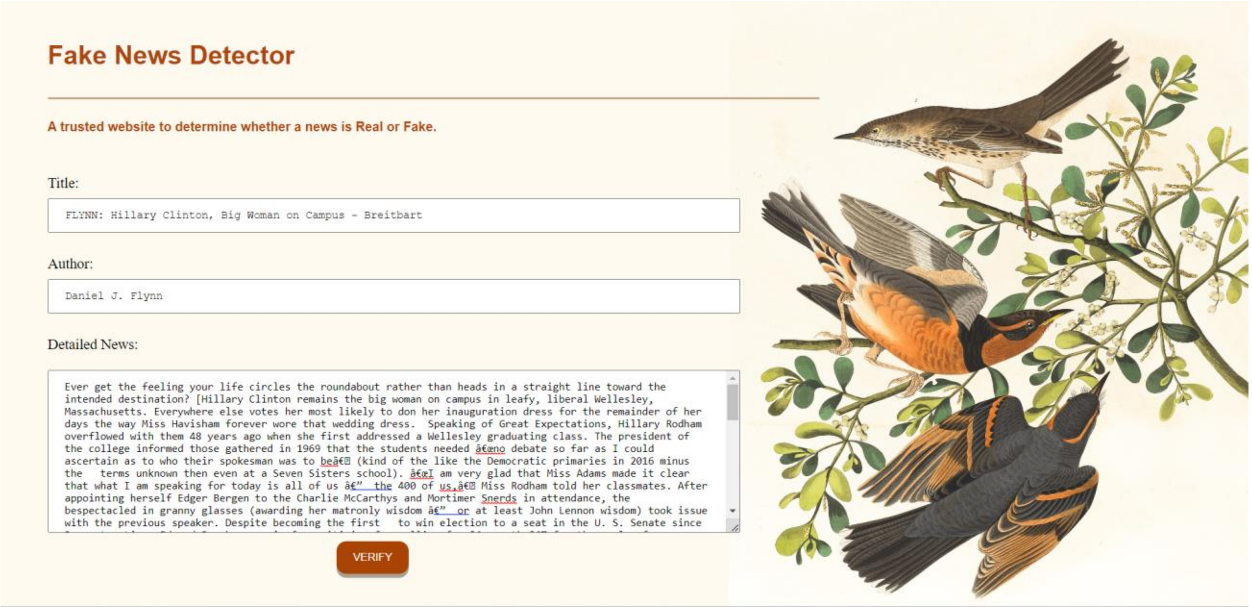
Author:

Type the author's name here...

Detailed News:

Type the detailed news here...

Fig. 4: Screenshot of Home Page



Fake News Detector

A trusted website to determine whether a news is Real or Fake.

Title:

FLYNN: Hillary Clinton, Big Woman on Campus - Breitbart

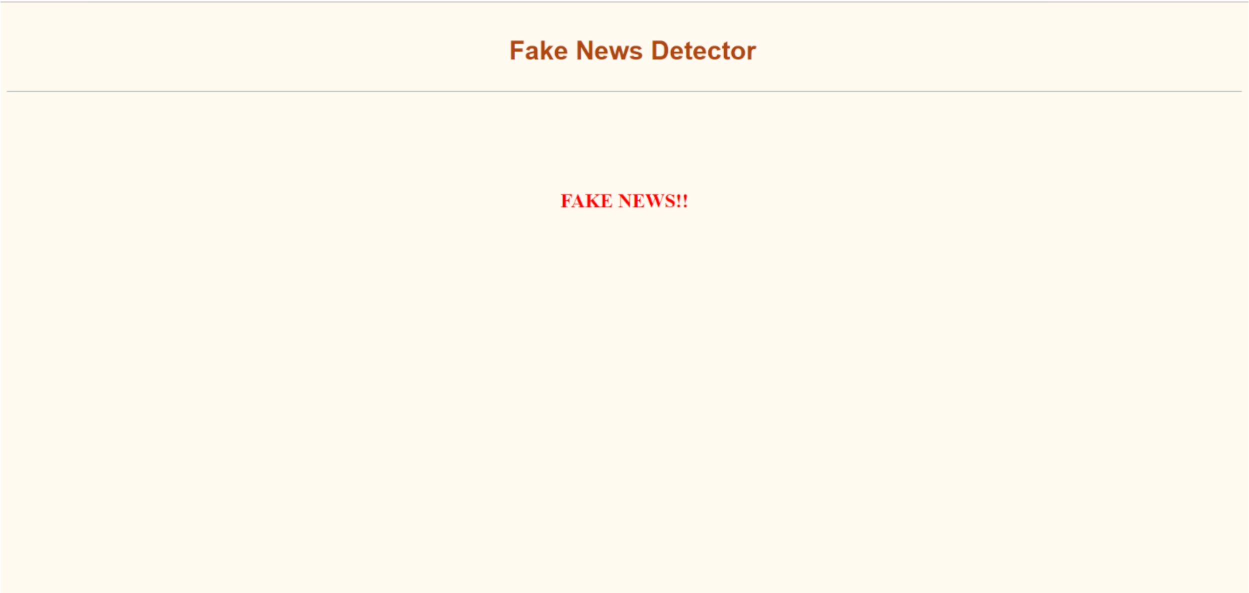
Author:

Daniel J. Flynn

Detailed News:

Ever get the feeling your life circles the roundabout rather than heads in a straight line toward the intended destination? [Hillary Clinton remains the big woman on campus in leafy, liberal Wellesley, Massachusetts. Everywhere else votes her most likely to don her inauguration dress for the remainder of her days the way Miss Havisham forever wore that wedding dress. Speaking of Great Expectations, Hillary Rodham overflowed with them 48 years ago when she first addressed a Wellesley graduating class. The president of the college informed those gathered in 1969 that the students needed â€œno debate so far as I could ascertain as to who their spokesman was to beâ€□ (kind of the like the Democratic primaries in 2016 minus the   terms unknown then even at a Seven Sisters school). â€œI am very glad that Miss Adams made it clear that what I am speaking for today is all of us â€" the 400 of us,â€□ Miss Rodham told her classmates. After appointing herself Edger Bergen to the Charlie McCarthys and Mortimer Snerds in attendance, the bespectacled in granny glasses (awarding her matronly wisdom â€" or at least John Lennon wisdom) took issue with the previous speaker. Despite becoming the first   to win election to a seat in the U. S. Senate since

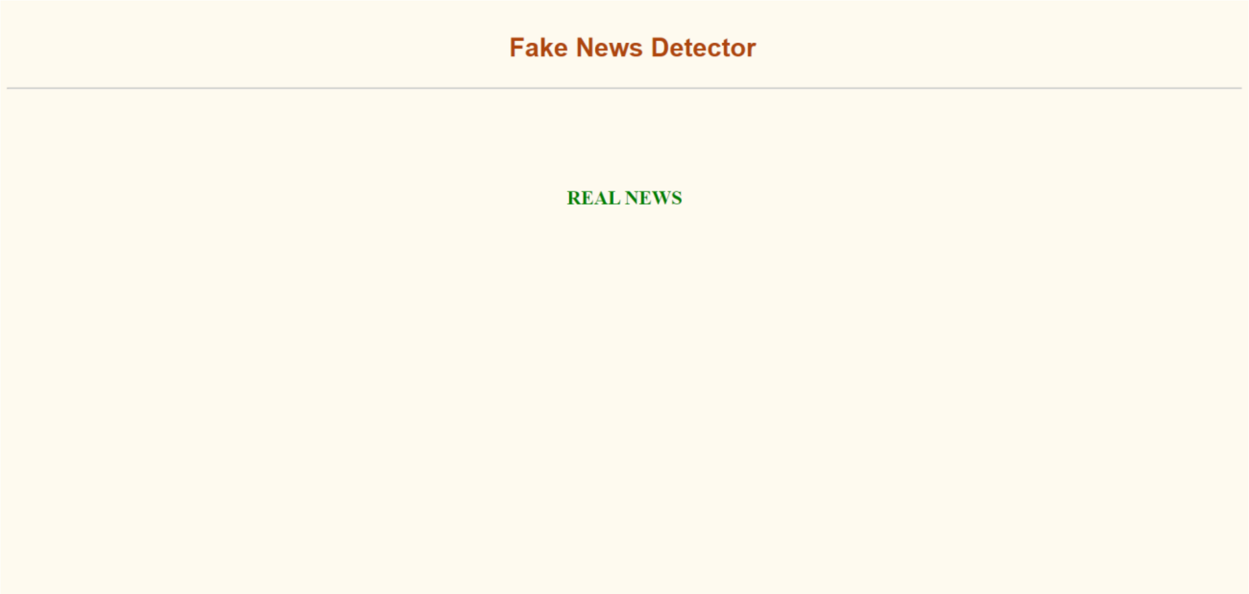
Fig. 5: Screenshot of Home Page after all inputs



Fake News Detector

FAKE NEWS!!

Fig. 6: Display of Fake News



Fake News Detector

REAL NEWS

Fig. 7: Display of Real News

6.3 Discussion and Result Analysis

The below figure summarizes the accuracy level of the Logistic Regression Machine Learning Model in the field of Fake News Detection as compared to other Machine Learning algorithms such as Random Forest, Decision Tree, Statistical Approach etc.

As observed below, the overall accuracy of the Logistic Regression algorithm is higher than other machine learning approaches. Hence, this fake News Detection system provides better accurate results as it is solely based on Logistic Regression Approach.

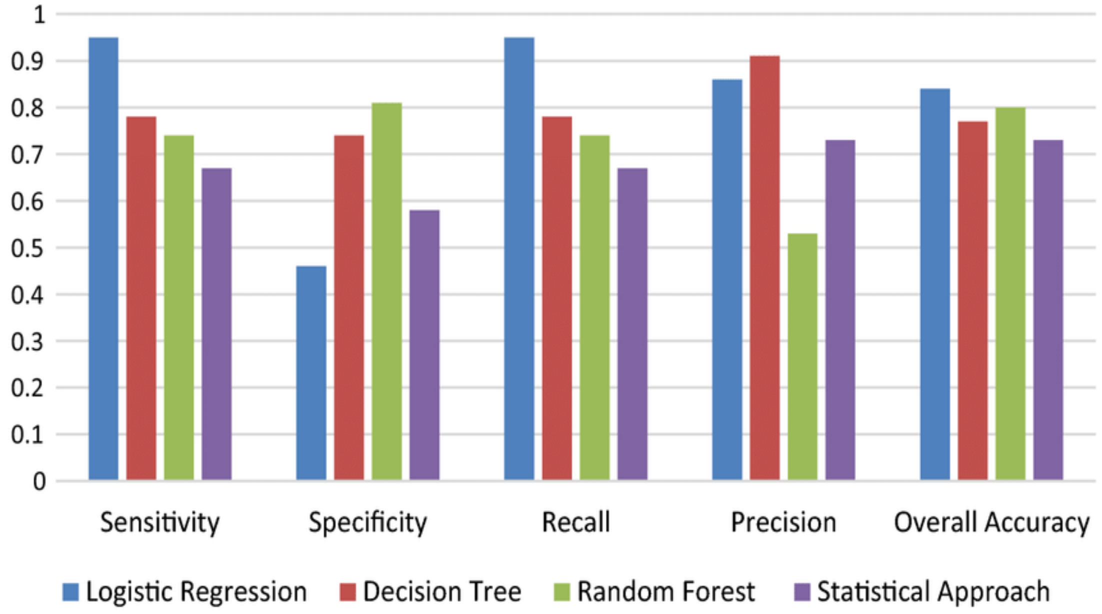


Fig. 8: Bar-chart comparing Logistic Regression and other ML Algorithms

Usage of NLTK package to remove stop words that are words that are used too often but hold less weightage such as ‘the’, ‘a’, ‘is’ etc. has been proved to increase the response time of the system along with ensuring better accuracy.

Below is the representation of frequent words after stop words removal from the news that are provided by the user to check its authenticity. The horizontal axis represents the words and the vertical axis indicates the number of times those words appeared during training, validation and testing phase.

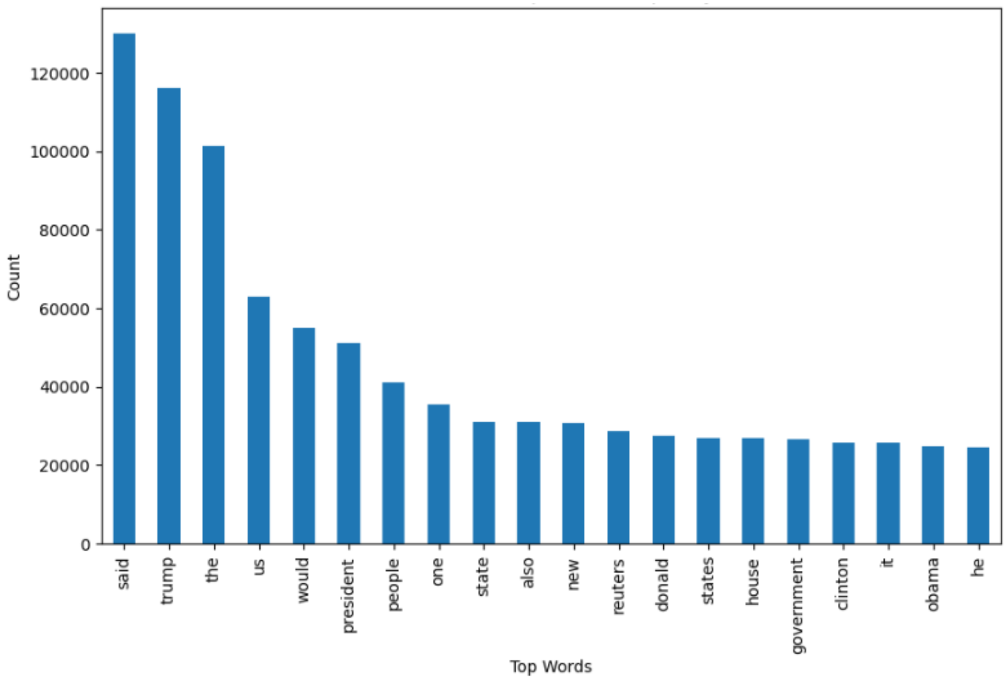


Fig. 9: Bar chart representation of stop words frequency

The datasets that are used during the training, validation and testing phase contain both fake and real news. After successfully training the model, the model is tested with unknown dataset to verify the performance of the aforementioned system.

The below figure represents the comparison of fake and real news that matches the expected and obtained result.

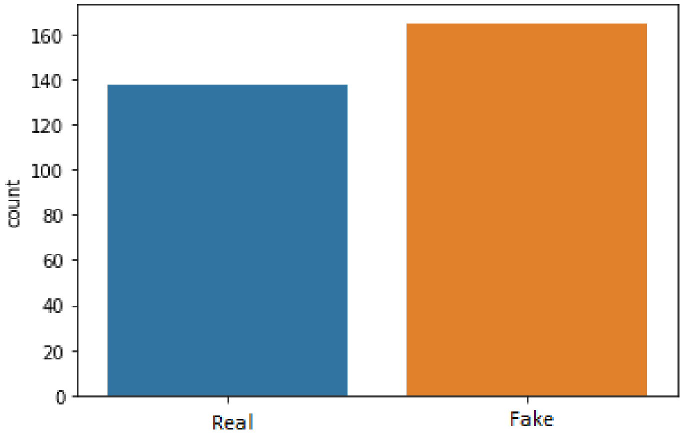


Fig. 10: Statistical Representation of Fake and Real news predicted by the System

After obtaining the predicted result from the testing dataset, evaluation is performed to analyze the accuracy and performance of the system. ROC Curve analysis, Confusion Matrix calculation etc. are performed for understanding of the obtained result which in turn conveyed the discrepancies between obtained and expected result as well.

Below figure represents the Confusion Matrix analysis on the labels obtained from testing dataset indicating True Positive, True Negative, False Positive and False Negative.

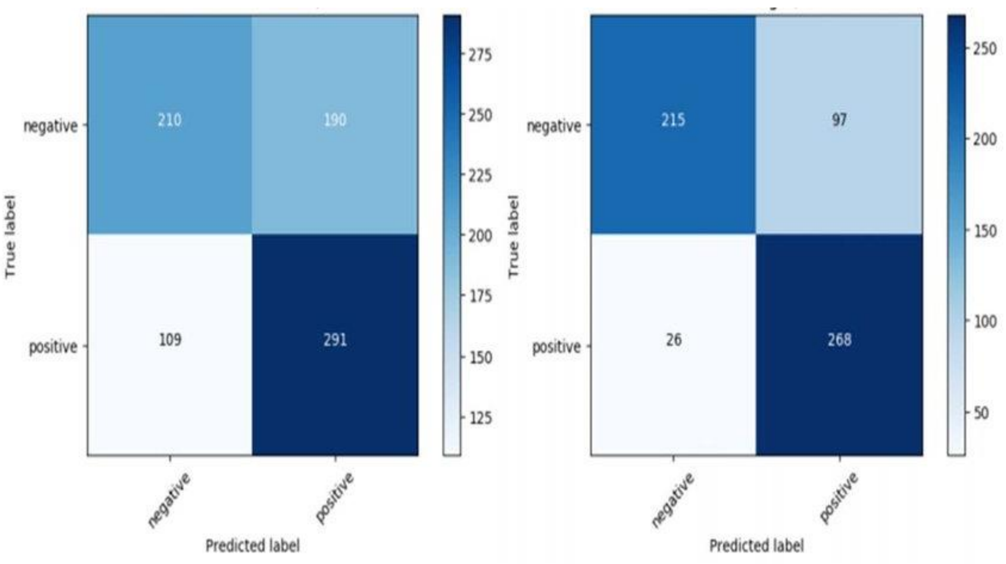
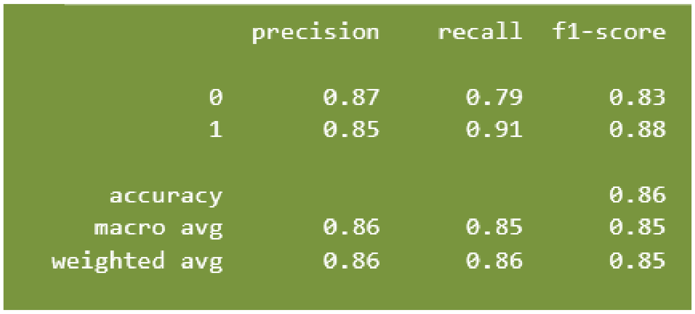


Fig. 11: Confusion Matrix Analysis

The final step of evaluation is to calculate the precision, recall and F1-score of the Logistic Regression Model to measure its accuracy.

Below figure represents the aforementioned parameters of model evaluation where 0 represents the label for Real news and 1 indicates that of Fake News.



```
              precision    recall  f1-score

           0       0.87      0.79      0.83
           1       0.85      0.91      0.88

    accuracy                           0.86
   macro avg       0.86      0.85      0.85
weighted avg       0.86      0.86      0.85
```

Fig. 12: Precision, Recall & F1-Score of the model used

Chapter 7: Conclusion and Future Work

7.1 Conclusion

Throughout the project, we embarked on a journey to develop an effective system for detecting fake news. Our focus was on building a website that would seamlessly integrate the logistic regression model with a user-friendly front end. Python served as the primary programming language for coding the website, and Flask, a powerful web framework, played a pivotal role in integrating the model with the front-end interface. Throughout the project, we explored various models, including the RNN model and passive aggressive model, to enhance our understanding of different approaches to fake news detection. Although these models were not utilized in the final implementation, they provided valuable insights into the complexities of the problem.

The challenges we encountered during the project spanned from preparing and cleaning the dataset to ensuring the accuracy and quality of the training and testing data. We employed a rigorous data preprocessing pipeline, which encompassed removing irrelevant information, handling missing data, and implementing text normalization techniques such as stemming and stop-word removal.

To evaluate the performance of our fake news detection system, we conducted extensive experiments, employing various evaluation metrics such as accuracy, precision, recall, and F1-score. The results of our analysis showcased the high accuracy of our system in effectively classifying news articles and detecting instances of fake news.

A significant achievement of this project was the successful deployment of the fake news detection website, made possible by integrating the logistic regression model with the Flask web framework. The website offered a user-friendly interface where users could input news articles and receive instant feedback on their authenticity. This integration involved implementing the necessary data processing and feature extraction pipelines to seamlessly connect the model with the front-end interface.

The successful completion of this project not only contributes to the field of fake news detection but also underscores the importance of technological solutions. With further research and development, the insights gained from this project can serve as a foundation for even more advanced and robust fake news detection systems in the future.

7.2 Future Work

The field of Fake News Detection continues to evolve, and there are several avenues for future development. They are as follows:

7.2.1 Expanding the Dataset:

To enhance the effectiveness and generalizability of the fake news detection system, expanding the dataset is crucial. A larger and more diverse dataset would provide a broader range of examples, allowing the model to learn from a wider variety of fake news instances. Incorporating different domains, languages, and cultural contexts would make the system more robust and applicable to real-world scenarios. Additionally, including more recent and up-to-date data would ensure that the system remains effective in detecting the latest trends and techniques used in spreading fake news.

7.2.2 Advanced Models:

While logistic regression has proven to be effective in detecting fake news, exploring advanced models can further enhance the system's performance. One such model is the Recurrent Neural Network (RNN), which has shown promise in modeling sequential data and capturing the contextual dependencies present in text. By leveraging the sequential nature of news articles, RNNs can potentially improve the system's ability to detect subtle patterns and linguistic nuances indicative of fake news.

7.2.3 Unsupervised Learning:

Incorporating unsupervised learning techniques can be beneficial for fake news detection. Unsupervised learning allows the system to identify patterns and anomalies in the data without relying on labeled examples. By leveraging clustering algorithms, anomaly detection methods, and topic modeling techniques, the system can uncover hidden patterns and characteristics associated with fake news articles. This approach can be particularly useful in scenarios where labeled data is scarce or expensive to obtain.

7.2.4 Reinforcement Learning:

Another promising direction for future work is the integration of reinforcement learning techniques into the fake news detection system. Reinforcement learning enables the system to learn through trial and error by interacting with its environment. By defining suitable rewards and penalties, the system can optimize its decision-making process and continuously improve its accuracy in detecting fake news. Reinforcement learning can also adapt to evolving trends and tactics employed by malicious actors in spreading misinformation, making the system more robust and resilient.

7.2.5 Multi-modal Approaches:

Expanding the fake news detection system to incorporate multi-modal data sources, such as images, videos, and social media content, can provide a more comprehensive understanding of the news articles. Combining textual information with visual cues and social context can enable the system to identify inconsistencies and discrepancies across different modalities, leading to more accurate detection of fake news.

Bibliography

Book Reference

[1] Kudari, Jayashree M., V. Varsha, B. G. Monica, and R. Archana. "Fake news detection using Passive aggressive and TF-IDF vectorizer." 2020 International Research Journal of Engineering and Technology (2020).

[2] Muhammad Syahmi Mokhtar, Yusmadi Yah Jusoh, NoviaAdmodisastro, NorainiChe Pa, AmruYusrinAmruddin, “Fakebuster: Fake News Detection System Using Logistic Regression Technique In Machine Learning ”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019.

[3] Sheng How Kong, Li Mei Tan, Keng Hoon Gan, Nur Hana Samsudin, “Fake News Detection using Deep Learning”, IEEE Xplore, June 06,2020.

[4] Fathima Nada, Bariya Firdous Khan, Aroofa Maryam, Nooruz-Zuha, Zameer Ahmed, “FAKE NEWS DETECTION USING LOGISTIC REGRESSION”, International Research Journal of Engineering and Technology (IRJET), May 2019

[5] Z Khanam , B N Alwasel , H Sirafi and M Rashid, “Fake News Detection Using Machine Learning Approaches”, IOP Conference Series: Materials Science and Engineering

[6] Dr.M. RAJESWARI1, A.SRINIRANJANI, “FAKE NEWS DETECTION USING LOGISTIC REGRESSION ALGORITHM WITH MACHINE LEARNING”, 2022 JETIR June 2022, Volume 9, Issue 6

[7] Dr.Rajiv Chopra, “Machine Learning”, 2nd edition, Learning Algorithms, Page No: 141, Khanna Book Publishing Co LTD, 2022

[8] Robert Picard, “Explore flask:Build, Deploy and Scale Python Web Applications”, 3rd edition, Page No:34, 129, 186

[9] Miguel Grinberg, “Flask Web Development with Python Tutorial”, 1st edition, Page No: 141

[10] Jiawei Han, Michelin Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, 2nd edition, Volume 2, Page No: 211, 232

[11] Ian H. Witten, Eibe Frank, Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", 1st edition, Page No:111, 132

[12] Salvador Garcia, Julian Luengo, Francisco Herrera, "Data Preprocessing in Data Mining", 3rd edition, Volume 2, Page No: 114, 211, 312

[13] Elizabeth Castro, Bruce Hyslop, "HTML & CSS:Visual QuickStart Guide", 4th edition

[14] Jennifer Niederst, "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics", 1st edition, Page No: 45, 56, 91

Web Reference

[15] Aman Kharwal, https://thecleverprogrammer.com/passive-aggressive-classifier-in-machine-learning/

[16] Susan Li, https://towardsdatascience.com/covid-fake-news-detection-with-a-very-simple-logistic-regression

[17] David Cournapeau, https://scikit-learn.org/stable/

[18] W3schools, https://www.w3schools.com/python/pandas/default.asp

[19] W3schools, https://www.w3schools.com/python/numpy/numpy_intro.asp

[20] Geeksforgeeks, https://www.geeksforgeeks.org/saving-a-machine-learning-model/

[21] Google researchers https://www.tensorflow.org/tutorials/keras/save_and_load

[22] W3schools, https://www.w3schools.com/html/

[23] Javatpoint, https://www.javatpoint.com/html-tutorial

[24] W3schools, https://www.w3schools.com/Css/

[25] Geeksforgeeks, https://www.geeksforgeeks.org/flask-tutorial/

[26] Tutorialspoint, https://www.tutorialspoint.com/flask/index.html