FAKE NEWS DETECTION USING FEATURE ENGINEERING IN MACHINE LEARNING

A Mini Project Report submitted to the Department of Computer Applications, Bharathiar University in the partial fullfillment of the requirements for the award of degree of

MASTER OF DATA ANALYTICS

Submitted by
KIRTHIKA V
(22CSEG16)

Under the guidance of

Dr. J. Komalalakshmi, B.Sc., MCA., M.Phil., B.Ed., M.Ed., Ph.D.,

GUEST FACULTY

DEPARTMENT OF COMPUTER APPLICATIONS

BHARATHIAR UNIVERSITY

COIMBATORE - 641 046

NOVEMBER – 2023

DECLARATION

I hereby declare that this Mini-project report titled **" FAKE NEWS DETECTION BY USING FEATURE ENGINEERING IN MACHINE LEARNING"** submitted to the Department of Computer Applications, Bharathiar University is a record of original work done by **KIRTHIKA V (22CSEG16)** under the guidance of **Dr. J. Komalalakshmi, B.Sc., MCA., M.Phil., B.Ed., M.Ed., Ph.D.,** Department of Computer Applications, Bharathiar University and this project work has not formed the basis for the award of any Degree/ Diploma/ Associate ship/ Fellowship or similar title to any candidate of any University.

Place: Coimbatore Signature of the Candidate

Date: (KIRTHIKA V)

Countersigned by,

Guide

CERTIFICATE

This is to certify that the Mini-Project report titled "**FAKE NEWS DETECTION USING FEATURE ENGINEERING IN MACHINE LEARNING"** submitted to the Department of Computer Applications, Bharathiar University in partial fulfilment of the requirement for the award of the degree of the Master of Computer Applications is record of the original work done by **KIRTHIKA V (22CSEG16)** under my supervision and guidance and this project work has not formed the basis for the award of any Degree/Diploma/Associate ship/ Fellowship or similar title to any candidate of any University.

Place: Coimbatore

Date:

Project Guide Head of the Department

Submitted for the University Viva-Voice Examination held on _____

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I express my respectful thanks to our Professor & Head of the Department **Dr. M. Punithavalli, B.Sc., M.Sc., M.Phil., Ph.D,** Department of Computer Applications, Bharathiar University, for permitting me to carry out my mini project report work in **"FAKE NEWS DETECTION USING FEATURE ENGINEERING IN MACHINE LEARNING"**

I really deem it a special privilege to convey my prodigious and everlasting thanks to my guide **Dr. J. Komalalakshmi, B.Sc., MCA., M.Phil., B.Ed., M.Ed., Ph.D.,** Department of Computer Applications, Bharathiar University, for her valuable guidance and personal interest in this mini project report work.

Last but not least, I also acknowledge the help done by my parents and acknowledge the encouraging support of my friends who were involved in this mini project, in one way or the other. I thank Almighty for showering the divine grace on me and offer prayers to the lord for everything that was given to me.

KIRTHIKA V (22CSEG16)

ABSTRACT

In a world flooded with information, it's becoming increasingly challenging to distinguish real news from fake news. To address this issue, this project employs the power of machine learning, specifically **Logistic regression**, to automatically identify whether a news article is genuine or fabricated. By collecting a diverse dataset of news articles, both real and fake. Then, using logistic regression, a popular machine learning algorithm, the model is developed that learns to recognize patterns and characteristics that distinguish real news from deceptive content.

The developed model is trained, and then integrate it into a user-friendly web application built with Streamlit. This platform allows users, even those without technical expertise, to interact with the model effortlessly. Simply input the text of a news article, and this application will swiftly determine its authenticity. This project aims to empower individuals with a tool that enhances media literacy and combats the spread of misinformation. Just a few clicks, anyone can verify the credibility of news articles, contributing to a more informed and discerning society.

System Development:

Backend:

In this project, the Scikit-Learn metrics library is used to develop a fake news detection model. The dataset was sourced from Kaggle and consisted of two main components: a training dataset and a test dataset.

Training Dataset	25,117 rows
Testing Dataset	5,881 rows
Total records	20, 799

These datasets were downloaded and stored as CSV files in local directory for model development and evaluation. This document summarizes the key aspects of the approach and dataset for the fake news detection project**.**

Frontend:

For the development of this fake news detection project, the Python programming language is employed as the primary tool. Within Python, harnessed the power of machine learning algorithms, with a particular focus on utilizing the Logistic Regression algorithm as the cornerstone of this project. This choice of algorithm played a crucial role in building an effective fake news detection model.

Connectivity:

The connection between Python and the Fake News Detection dataset has been established, and subsequently, the dataset will imported into the Python environment.

(i) pd.read_csv("train.csv")
(ii) pd.read_csv("test.csv")

This was accomplished by utilizing the Pandas library, employing the pd.read_csv function to read and load the dataset from the CSV file into the Python environment for further analysis and model development.

Work Flow:

The Fake News Detection dataset includes Seven Features which are

(i) ID
(ii) Title
(iii) Author
(iv) Test
(v) Labels
(vi) Subject
(vii) Date

Among the Seven Features the following Two key features namely,

(i) Title
(ii) Labels

have been selected for predicting this model.

Out of the total 20,799 records, two news articles were deliberately selected:

(i) One of the selected news articles, classified as fake news, corresponds to the index number 20806 and is available in row number 8.

(ii) The other news article, classified as true news, corresponds to the index number 20811 and is available in row number 13.

These two articles were chosen to evaluate the model's classification accuracy, determining whether it can correctly classify each article into its respective category.

Web development:

Streamlit is a Python library and open-source framework that is specifically designed for rapid and interactive web application development. It serves as an ideal choice for web development due to its user-friendly and intuitive approach, allowing developers to effortlessly transform data scripts into web applications with minimal effort.

The Set path for Streamlit Web Development:

C:\user\Chandru\Documents\Python code\Projects\Fake News.py

In this project, the power of Streamlit is used to create a user-friendly web interface for the fake news detection model. Streamlit's ease of use and flexibility allowed to quickly transform this machine learning model into an interactive web application, making it accessible and user-friendly for a wide audience.

In the future, the model can be expanded to include a larger volume of offline records with more features incorporating data from online website news and app news sources.

TABLE OF CONTENTS

I. INTRODUCTION

Imagine this: every day, we come across lots of news on the internet. Some of it is true, while some of it is made up – fake news. Fake news is like a trick; it pretends to be real, but it's not. This can cause problems because it confuses people and can even make them believe things that aren't true.

This project is like a superhero trying to stop fake news. By using the power of computers and smart technology to figure out if a piece of news is real or fake. It's like a lie detector for news!

By building a special computer program using something called "machine learning." This program learned from lots of news stories to become really good at spotting fake ones. And the best part is, don't need to be a computer expert to use it. To made it super easy for anyone to check if a news story is telling the truth.

By doing this, hope to make the internet a better place where people can trust the news they read. This project is like a friend you can count on to help you know what's true and what's not.

So, let's dive in and see how this project works and how it can make a big difference in the world of news and information!

This introduction simplifies the project's purpose and approach, making it easy to understand for a wide audience.

1.1 BACKGROUND AND MOTIVATION

- **Background:**

In today's digital age, the ease of sharing and accessing information has reached unprecedented levels. The rapid dissemination of news and information through the internet and social media platforms has revolutionized how to stay informed. However, this rapid information flow has also given rise to a pressing issue - the proliferation of fake news and misinformation.

Fake news, often presented as credible and accurate reporting, can have severe consequences for individuals, society, and even the political landscape. It can incite panic, sway public opinion, and erode trust in journalism and information sources. Detecting and combatting fake news has become a crucial endeavor in the modern world.

- **Motivation:**

The motivation behind this project stems from the growing need to tackle the fake news problem. Our society's increasing reliance on digital platforms for news consumption has made it susceptible to the spread of false information. The consequences of this misinformation can be far-reaching, affecting public health, political discourse, and social cohesion.

This project aims to address the following key motivations:

(i) **Media Literacy:** Enhancing media literacy is critical in the era of information overload. By providing a tool for users to discern real news from fake news, it empower them to make informed decisions about the information they consume and share.

(ii) **Misinformation Mitigation:** Misinformation can lead to real-world harm, such as the spread of false health advice, the distortion of political debates, or the incitement of violence. Our project's goal is to contribute to the reduction of such harm by identifying and flagging deceptive content.

- **Trust Restoration:** The credibility of news sources has come into question due to the proliferation of fake news. Restoring trust in journalism and credible reporting is an important societal goal, and this project supports that aim.
- **Technological Advancement:** Leveraging machine learning and natural language processing to detect fake news is an example of harnessing technology for the common good. The project serves as a testament to the innovative applications of these fields.
- **User Empowerment:** By converting the fake news detection model into a user-friendly Streamlit application, ensure that the benefits of this research are accessible to a wide audience. This promotes active user participation in the fight against misinformation.

1.2 OBJECTIVE

The objective of this project is to examine the problems and possible significances related with the spread of fake news. By working on different fake news data set in which will apply different machine learning algorithms to train the data and test it to find which news is the real news or which one is the fake news. As the fake news is a problem that is heavily affecting society and our perception of not only the media but also facts and opinions themselves. By using the artificial intelligence and the machine learning, the problem can be solved to mine the patterns from the data to maximize well defined objectives. So, the focus is to find which machine learning algorithm is best suitable for what kind of text dataset. Also, which dataset is better for finding the accuracies, as the accuracies directly depend on the type of data and the amount of data. The more the data, more are chances of getting correct accuracy as can test and train more data to find out the results

1.3 OVERVIEW OF THE PROJECT

With the advancement of technology, digital news is more widely exposed to users globally and contributes to the increment of spreading and disinformation online. Fake news can be found through popular platforms such as social media and the Internet. There have been multiple solutions and efforts in the detection of fake news where it even works with tools. However, fake news intends to convince the reader to believe false information which deems these articles difficult to perceive. The rate of producing digital news is large and quick, running daily at every second, thus it is challenging for machine learning to effectively detect fake news

II. DATA COLLECTION AND PREPROCESSING

2.1 Data Sources

- **Specialized Datasets:**

There are specialized datasets curated for the purpose of fake news detection. These datasets may include both real and fake news articles. Examples include the "Fake News Challenge" dataset. These datasets are useful for training and testing machine learning models.

2.2 Data cleaning and Preprocessing

- **Data Cleaning:**
  - (i) **Handling Special Characters:** News articles may contain special characters, punctuation, or symbols. These can be removed or replaced with spaces to avoid affecting the analysis.
  - (ii) **Dealing with Noise**: Inconsistent capitalization, typos, or grammatical errors can introduce noise into the data. Cleaning involves standardizing text to improve analysis accuracy**.**
- **Text Tokenization:**
  - (i) **What is Tokenization:** Tokenization breaks the text into smaller units, typically words or phrases (tokens). It is a fundamental step in natural language processing.
  - (ii) **Importance:** Tokenization allows the model to understand the text structure, making it easier to analyze and process.
- **Stopword Removal:**
  - (i) **Identifying Stopwords**: Stopwords are common, non-informative words such as "the," "and," and "is." Removing them reduces data dimensionality.
  - (ii) **Why Remove Stopwords:** Stopwords don't carry much meaning and can be safely discarded without affecting the analysis.
- **Lowercasing:**
  - (i) **Standardizing Text:** Converting all text to lowercase ensures that the model treats words with different cases as the same.
  - (ii) **Consistency:** This step ensures that the model focuses on the meaning of words rather than their capitalization.

- **Stemming or Lemmatization:**
  (i) **What is Stemming/Lemmatization:** Stemming reduces words to their base form, while lemmatization considers the word's root (lemma)
  (ii) **Importance**: It ensures that different variations of the same word are treated as one. It can improve the model's ability to recognize related words.
- **Text Vectorization:**
  (i) **Converting Text to Numbers:** Machine learning models work with numerical data. Text data must be transformed into numerical vectors. Common methods include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings (e.g., Word2Vec or GloVe).
  (ii) **Word Embeddings:** Word embeddings map words to numerical vectors, capturing semantic relationships between words. These vectors help the model understand the context and meaning of words.
- **Handling Imbalanced Data:**

Techniques like oversampling (creating more fake news examples) or undersampling (reducing real news examples) can address this imbalance and help the model learn effectively.

- **Data Splitting:**

Divide the dataset into three parts: the training set (for teaching the model), the validation set (for fine-tuning), and the test set (for evaluating the model). This ensures that the model's performance is assessed on unseen data.

2.3 Labelling and Categorization

- **Labeling:**
  (i) **Defining Real and Fake News:** In the context of this project, labeling involves categorizing news articles as either "real" or "fake." Real news is factual and accurate, while fake news is intentionally misleading or false.
  (ii) **Manual vs. Automated Labeling:** In many cases, this process is manual, where human annotators read each article and decide its category. However, this can also employ automated techniques or pre-labeled datasets.

- **Data Balancing:**

(i) **Imbalance Considerations:** The real and fake news articles may not be evenly distributed in the dataset. There might be more real news than fake news or vice versa.

(ii) **Addressing Imbalance:** The techniques that is used to balance the dataset. This can include oversampling (creating more fake news examples), undersampling (reducing real news examples), or a combination of both to ensure fairness in model training.

- **Categorization:**

The various categories of fake news, such as satire, misinformation, clickbait, or propaganda. This project includes the categorization of fake news and misinformation.

III. FEATURE ENGINEERING

Feature engineering is a crucial step in the process of preparing data for machine learning models, and it plays a significant role in this project to detect fake news.

- **What is Feature Engineering?**

Feature engineering is the process of creating new features or transforming existing features in the dataset to make it more suitable for machine learning algorithms. These features are the characteristics or properties of the data that the machine learning model uses to make predictions. Feature engineering involves selecting, transforming, and creating features to improve the model's performance and accuracy.

- **Importance of Feature Engineering:**

Feature engineering is a fundamental step in the machine learning pipeline because it directly impacts the model's ability to learn and make predictions. Well-engineered features can lead to a more accurate and efficient model, while poorly chosen or unprocessed features can hinder model performance.

- **Feature Engineering in Fake News Detection:**

In the context of this project to detect fake news, feature engineering is particularly important because it helps the model understand and distinguish between real and fake news articles. Here are some common feature engineering techniques and considerations for this project:

(i) **Text Representation:** The primary data in this project is the text of news articles. It will need to convert this text into numerical representations that machine learning models can work with.

- **Common methods include:**

(i) **TF-IDF (Term Frequency-Inverse Document Frequency):** This technique measures the importance of words in a document relative to a collection of documents. It assigns a weight to each word based on its frequency in the document and its rarity in the entire collection.

(ii) **Word Embeddings:** Word embeddings like Word2Vec or GloVe are pre-trained models that represent words as dense vectors, capturing semantic relationships between words.

3.2 Feature Extraction

Easily can extract various features from the text data to help the model differentiate between real and fake news. Examples of extracted features might include:

- **Word Count:** The number of words in the Dataset
- **Sentence Count:** The number of sentences in the Dataset
- **Readability Metrics:** Metrics like the Flesch-Kincaid Grade Level or Coleman-Liau Index that assess the complexity of the text.
- **N-grams:** Pairs or sequences of adjacent words in the text.

3.3 Handling Imbalanced Data

Handling imbalanced data is a crucial consideration in machine learning, especially in tasks like fake news detection where one class (e.g., real news) significantly outweighs the other (e.g., fake news) in terms of the number of samples.

- **Understanding the Imbalance:**
  (i) **Class Distribution:** Imbalanced data occurs when one class has significantly fewer samples than the other. For example, in fake news detection, it has many more real news articles than fake ones.
  (ii) **Impact:** Imbalance can lead to biased models. A machine learning model trained on imbalanced data may become overly biased toward the majority class, resulting in poor detection of the minority class (e.g., fake news).
- **Techniques for Handling Imbalanced Data:**

There are several strategies to address imbalanced data in the fake news detection project:

(i) **Oversampling:** Increase the number of samples in the minority class. This can be done by duplicating existing samples or generating synthetic examples.

(ii) **Undersampling:** Decrease the number of samples in the majority class by randomly removing samples. This method can balance the class distribution but may lead to loss of information.

- **Weighted Loss:** Assign higher weights to the minority class during model training. This way, the model pays more attention to the minority class samples and tries to minimize errors for that class.
- **Generate Synthetic Data:** Techniques like Synthetic Minority Over-sampling Technique (SMOTE) create synthetic examples of the minority class based on the existing samples. This expands the dataset with artificial but realistic examples.
- **Anomaly Detection:** Consider treating the minority class as an anomaly detection problem, to identify unusual instances as fake news. This approach involves finding deviations from the majority class's distribution.
- **Ensemble Methods**: Ensemble techniques like Random Forest or AdaBoost can be effective in handling imbalanced data by combining multiple models. They can give more weight to the minority class, improving detection.
- **Cost-Sensitive Learning:** Adjust the model's learning algorithm to be cost-sensitive, meaning it penalizes errors differently for each class. Misclassifying the minority class is penalized more heavily.

IV. MACHINE LEARNING MODEL (LOGISTIC REGRESSION)

Logistic regression is a popular machine learning model used for classification tasks. It's particularly useful when you want to predict one of two possible outcomes (binary classification) like whether the news article is fake or not.

4.1 Introduction to Logistic Regression

In the pursuit of detecting fake news, the role of machine learning models cannot be overstated. These models act as vigilant gatekeepers, sifting through vast volumes of information to distinguish fact from fiction. One such model, known as logistic regression, plays a pivotal role in this endeavor.

- **Binary Classification:** Logistic regression is a foundational machine learning algorithm specifically designed for binary classification tasks. In binary classification, the goal is to predict one of two possible outcomes, often represented as "0" or "1," "True" or "False," or, in our case, "Real" or "Fake."
- **Modeling Probabilities:** It sets logistic regression apart is its ability to model probabilities. Rather than directly classifying an input as either real or fake, it estimates the probability that an input belongs to the positive class (fake news) or the negative class (real news). This probability estimation is a critical element in fake news detection.
- **Logit Function:** The magic behind logistic regression lies in the logit function, which transforms a linear combination of features into a probability score. This function, also known as the sigmoid or logistic function, produces an "S"-shaped curve, smoothly transitioning from 0 to 1. The value it yields represents the likelihood of an input being associated with fake news.

4.2 Model Architecture

In the realm of fake news detection, the model architecture serves as the heart and brain of the system, responsible for distinguishing between authentic and deceptive news articles. The architecture dictates how the model processes data, makes predictions, and ultimately contributes to the project’s success.

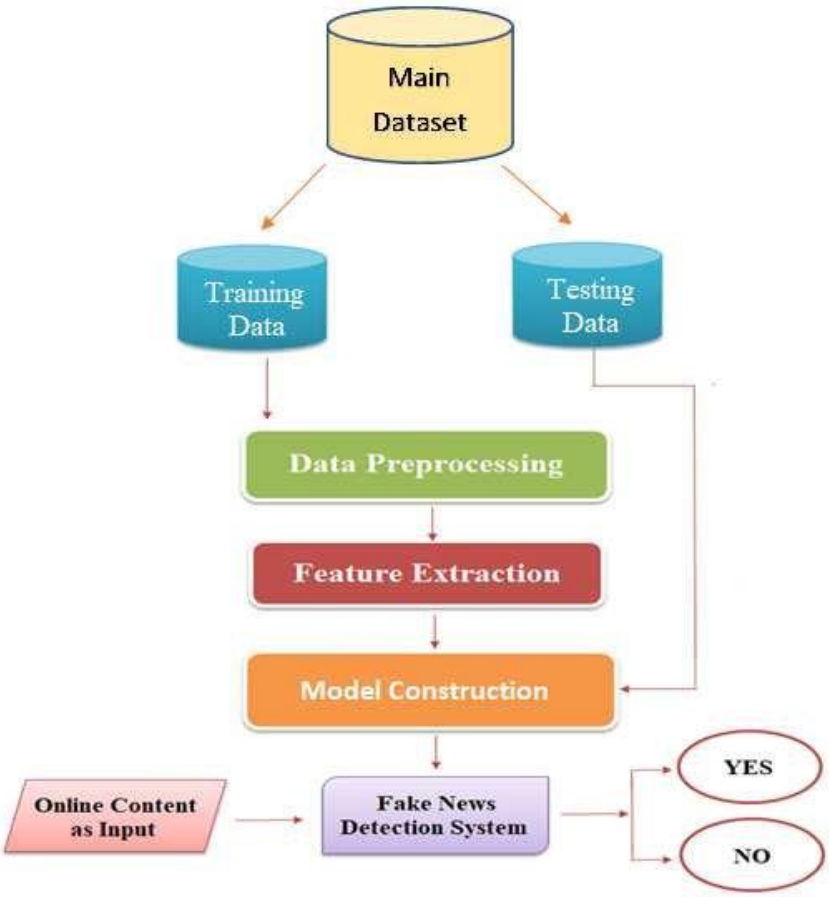


Fig 4.1 Model Architecture Diagram

- **Input Layer**:

At the base of the model architecture is the input layer, which receives the data to be analyzed. In the case of fake news detection, this typically involves the text content of news articles. The input layer is responsible for receiving and encoding the text data for further processing.

- **Feature Engineering:**

Before the data enters the model, it often undergoes feature engineering, which involves extracting relevant features or characteristics from the text. These features can include word frequencies, sentiment scores, topic information, and more. Feature engineering plays a crucial role in translating the raw text into a format that the model can work with.

- **Hidden Layers:**

The core of the model architecture lies in its hidden layers, which perform the actual data analysis. The number of hidden layers and the number of neurons (units) in each layer can vary based on the complexity of the problem.

- **Output Layer:**

The output layer of the model architecture is where the final classification decision is made. In fake news detection, this is a binary classification task. The output layer produces a probability score, typically using the sigmoid activation function, which is then compared to a threshold (e.g., 0.5) to classify the input as real or fake.

4.3 Model Justification

Model justification is an essential aspect of this project documentation in the context of fake news detection. It involves explaining particular machine learning model for this task and justifying its suitability.

- **Problem Suitability:**

Logistic regression is suitable for binary classification tasks, making it ideal for distinguishing between real and fake news.

- **Interpretable Results:**

Logistic regression provides easily interpretable results. It assigns weights (coefficients) to features, allowing to identify the most influential factors in classifying news articles. This transparency can be critical in understanding and explaining the model's decisions, a valuable trait when dealing with news credibility.

- **Feature Importance**:

The importance of feature engineering in fake news detection is that logistic regression effectively utilizes these engineered features, helping the model understand the distinctive characteristics of fake and real news articles.

- **Training Efficiency:**

Logistic regression is computationally efficient, making it suitable for processing large volumes of news articles. Its simplicity often results in faster training times compared to more complex models, which can be beneficial for real-time or batch processing.

- **Ease of Implementation:**

Logistic regression is relatively easy to implement and interpret. It doesn't require extensive hyperparameter tuning or complex architecture design, making it an approachable choice for practitioners.

- **Performance Metrics:**

The performance metrics are used to assess the model's effectiveness in fake news detection.

V. WEB APPLICATION

Streamlit is a popular Python library that allows you to create web applications with minimal effort. It's an excellent tool for visualizing and sharing the fake news detection project results.

- **What is Streamlit?**

Streamlit is an open-source Python library that simplifies the process of creating web applications. It is designed to be user-friendly, making it accessible to data scientists, developers, and domain experts who may not have extensive web development experience.

- **Interactivity and Visualization:**

Streamlit enables to turn the machine learning models and data analysis into interactive web applications. It's a versatile tool for presenting the results of fake news detection project, allowing users to interact with the data and model outputs.

- **Easy-to-Use:**

One of the key advantages of Streamlit is its simplicity. It can create a basic web application with just a few lines of Python code. This ease of use makes it an excellent choice for rapidly prototyping and sharing this project with a wider audience.

- **Data Integration:**

Streamlit seamlessly integrates with various data sources, enabling to display the fake news detection results and data visualizations within the web application. It can also connect to databases, APIs, and external data to provide up-to-date information.

- **Building the User Interface:**

Building a user interface for this project documentation involves creating a visually appealing and user-friendly design that allows users to interact with this project's features and findings. This user interface is often integrated into a web application using tools like Streamlit or other web development frameworks.

- **Purpose of the User Interface:**

The user interface is designed to provide an interactive and accessible platform for users to engage with the fake news detection model and view project findings.

- **Interactive Elements:**

The interactive elements incorporated into the user interface. These can include input fields for users to enter news articles, buttons to trigger predictions, and graphical representations of project results.

- **Input Features:**

Users can input data into the user interface. In the context of fake news detection, this typically involves allowing users to enter or paste text from news articles that they want to analyze.

- **Prediction Output:**

The user interface displays prediction results. When a user submits a news article, the model's prediction (real or fake) is presented, and whether it includes probability scores or confidence levels.

- **Visualizations and Insights:**

This project includes data visualizations or insights, these are integrated into the user interface. Users can interact with charts, tables, or visual representations of this project findings.

- **User-Friendly Design:**

Emphasize that it is designed to be user-friendly, with clear and intuitive navigation and a visually pleasing layout.

- **Responsive Design:**

The user interface is responsive, meaning it can adapt to different screen sizes and devices. This ensures that users can access your project from various platforms, including desktops, tablets, and smartphones.

- **Real-Time Updates:**

If the user interface supports real-time updates, users can also receive the latest information or predictions as this project evolves.

- **Error Handling and Feedback:**

The user interface handles errors or unexpected inputs from users, whether there are feedback mechanisms to guide users in case of errors.

- **Accessibility:**

The user interface is accessible to all users, including those with disabilities. This may include features like text-to-speech functionality or keyboard navigation.

- **Integration with the Model:**

The user interface interacts with this fake news detection model. Users should understand that the interface serves as a frontend to the machine learning model, allowing them to utilize its capabilities.

- **Deployment and Accessibility:**

Users can access the user interface. Whether it's hosted on a website, available as a standalone application, or integrated into the project documentation, provide clear instructions for users to access and navigate the interface.

- **User Guidance:**

Offer guidance or tooltips within the user interface to help users understand how to use it effectively. This can include instructions on input requirements or explanations of displayed results.

5.2Integrating the Logistic Regression Model

Integrating a logistic regression model into this project documentation is a critical step in explaining how the model is applied to detect fake news. This integration should provide a clear and comprehensive understanding of how the model works and how it contributes to the project's objectives

- **Purpose and Context:**

By providing context for the integration of the logistic regression model. This model is crucial for this fake news detection project and it fits into the overall framework.

- **Data Preparation:**

The data preprocessing steps that are necessary before integrating the logistic regression model. This can include text cleaning, feature extraction, and data encoding to prepare the input data for the model.

- **Feature Engineering:**

Feature engineering is a fundamental component of integrating the logistic regression model. The features used in the model, such as word frequencies, sentiment analysis scores, or any other relevant characteristics derived from the text data.

- **Model Integration:**

The integration of the logistic regression model shows that the prepared data is fed into the model and predictions are generated.

- **Prediction Output:**

The model generates binary predictions, typically labeled as "Real" or "Fake," and may also provide probability scores for the predicted class.

- **Threshold Setting:**

The decision threshold such as 0.5, is chosen to determine whether a news article is classified as real or fake based on the predicted probability.

- **Explainable Results:**

The model assigns weights (coefficients) to features, allowing users to understand which features influence its decisions.

- **Visualization of Predictions:**

If integrated the logistic regression model into a user interface or web application, the model's predictions are presented to users. This may involve visual elements, such as charts or color-coded indicators, to convey prediction results.

- **Performance Metrics:**

The use of Performance metrics like accuracy, precision, recall, F1 score, and AUC-ROC to assess how well the logistic regression model classifies fake and real news articles.

- **Handling Imbalanced Data:**

The applied strategies for handling imbalanced data (such as resampling or weighted loss), these techniques were integrated into the logistic regression model to improve its effectiveness in the presence of imbalanced classes.

- **Model Interpretation:**

The logistic regression model's coefficients allow users to understand which words or features contribute to a news article being classified as fake or real.

- **Continuous Monitoring and Updates:**

The logistic regression model can be continuously monitored and updated as new data becomes available, allowing it to adapt to evolving fake news patterns.

- **User Guidance:**

Provide guidance for users on how to interpret the model's predictions, including what to look for in the output and how to utilize the results effectively.

- **Model Integration into User Interface:**

The logistic regression model is integrated into this web application or user interface, allowing users to interact with the model directly.

- **Real time predictions**

Integrating real-time predictions into this Streamlit web application is a valuable feature that enhances the user experience and provides immediate feedback to users as they interact with the fake news detection system.

- **Objective and Significance:**

The objective of incorporating real-time predictions into the Streamlit web application. Emphasize the significance of this feature in providing users with instant feedback on the authenticity of news articles.

- **Model Integration:**

The logistic regression model is seamlessly integrated into the Streamlit web application. Users should understand that the model is part of the application, allowing it to provide predictions in real time.

- **User Input:**

Users can input news articles or text data into the Streamlit web application. Highlight any user-friendly input mechanisms you've implemented, such as text boxes or file upload options.

- **Predict Button:**

The user interface element (e.g., a button) that users can click to trigger predictions. When the button is pressed, it signals the application to send the user's input to the logistic regression model for analysis.

- **Backend Processing:**

The backend processing occurs when a prediction request is initiated. The Streamlit application processes the user's input, prepares it for analysis, and sends it to the logistic regression model.

- **Real-Time Feedback:**

Emphasize that as soon as the logistic regression model receives the input data, it performs real-time predictions. The results are then displayed in the application, allowing users to see the model's classification immediately.

- **Prediction Output:**

The prediction output is presented to users in real time. This may include labeling the news article as "Real" or "Fake," along with associated probability scores or confidence levels.

- **Threshold Setting:**

If applicable, the model uses a predefined threshold to determine the classification (e.g., real or fake). Users should understand how the decision boundary is applied to the real-time predictions.

- **Continuous Monitoring:**

The logistic regression model can be continuously monitored for updates. This ensures that the model remains effective in detecting evolving fake news patterns.

- **Visual Feedback:**

Visual elements that provide feedback to users. These may include color-coded indicators, progress bars, or chart representations of prediction results to enhance the user experience.

- **User Guidance:**

Offer guidance to users on how to interpret and act upon the real-time predictions. What to look for in the output and how they can use this information to make informed decisions about the credibility of news articles.

- **Integration with Other Features:**

If the Streamlit web application includes other features, such as data visualizations or insights, these elements complement the real-time predictions and provide a comprehensive understanding of the data.

- **Performance Metrics:**

The application provides performance metrics (e.g., accuracy, precision, recall) for the logistic regression model's predictions, allowing users to assess the model's effectiveness.

- **Use Cases and Benefits:**

The use cases and benefits of real-time predictions in this Streamlit web application, highlighting how this feature contributes to the project's goals and user engagement.

VI. RESULTS AND DISCUSSIONS

6.1Software Environment

- **PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter.

No need to compile the program before executing it. This is similar to

PERL and PHP.

Python is Interactive − Actually sit at a Python prompt and interact

with the interpreter directly to write the programs.

Python is Object-Oriented − Python supports Object-Oriented style or

technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the

beginner-level programmers and supports the development of a wide range of

applications from simple text processing to WWW browsers to games.

- **History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties

at the National Research Institute for Mathematics and Computer Science in the

Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

➢ PYTHON FEATURES

Python's features include:

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − Add low-level modules to the Python interpreter.

These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below

(i) It supports functional and structured programming methods as well as OOP.

(ii) It can be used as a scripting language or can be compiled to byte-code for building large applications.

(iii) It provides very high-level dynamic data types and supports dynamic type checking.

(iv) It supports automatic garbage collection.

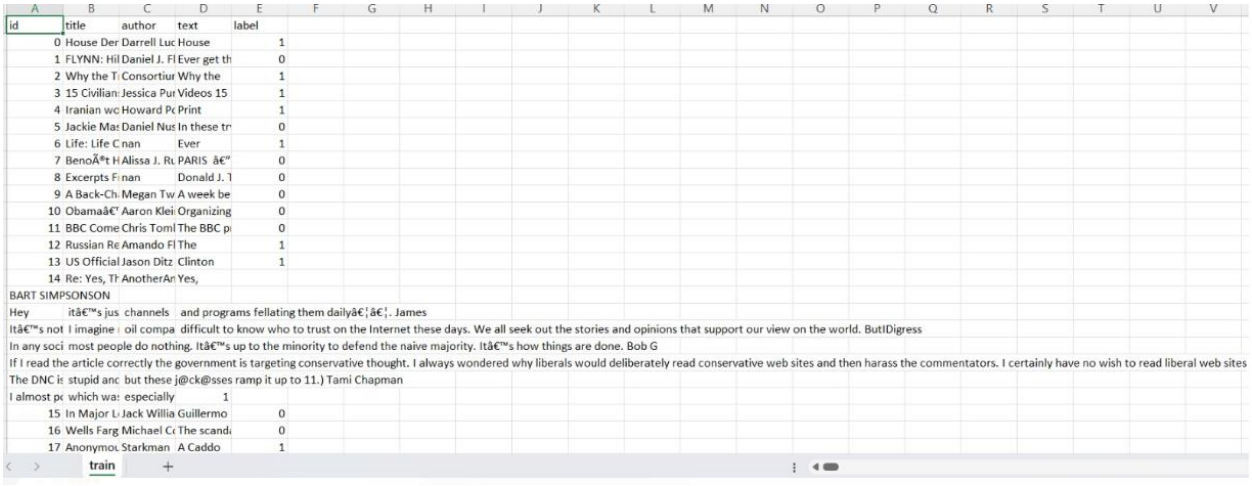
- **Script mode programming**

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Importing Necessary libraries for execute this project and then Load the dataset. After loading the dataset it is necessary to preprocess the dataset.

- **Dataset**

#Training dataset

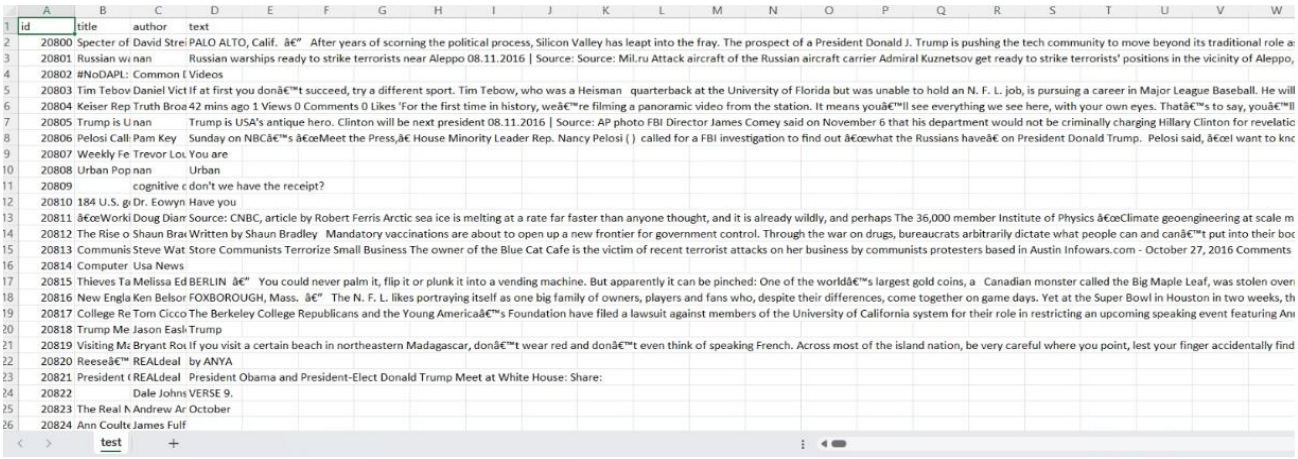


id	title	author	text	label
0	House Der	Darrell Luc	House	1
1	FLYNN: Hil	Daniel J. Fl	Ever get th	0
2	Why the T	Consortiur	Why the	1
3	15 Civilian	Jessica Pur	Videos 15	1
4	Iranian wc	Howard Pc	Print	1
5	Jackie Mas	Daniel Nus	In these tr	0
6	Life: Life C	nan	Ever	1
7	BenoÃ®t H	Alissa J. Rt	PARIS â€"	0
8	Excerpts F	nan	Donald J. T	0
9	A Back-Ch	Megan Tw	A week be	0
10	Obamaâ€™	Aaron Klei	Organizing	0
11	BBC Come	Chris Toml	The BBC p	0
12	Russian Re	Amando Fl	The	1
13	US Official	Jason Ditz	Clinton	1
14	Re: Yes, Th	AnotherAn	Yes,	
BART SIMPSONSON				
Hey	itâ€™s jus	channels	and programs fellating them dailyâ€¦â€¦. James	
Itâ€™s not	I imagine	oil compa	difficult to know who to trust on the Internet these days. We all seek out the stories and opinions that support our view on the world. ButIDigress	
In any soci	most people do nothing. Itâ€™s up to the minority to defend the naive majority. Itâ€™s how things are done. Bob G			
If I read the article correctly the government is targeting conservative thought. I always wondered why liberals would deliberately read conservative web sites and then harass the commentators. I certainly have no wish to read liberal web sites				
The DNC is	stupid anc	but these j@ck@sses ramp it up to 11.) Tami Chapman		
I almost pc	which wa:	especially	1	
15	In Major L	Jack Willia	Guillermo	0
16	Wells Farg	Michael Ce	The scand	0
17	Anonymou	Starkman	A Caddo	1

train

Fig 6.1 Training dataset

#Testing dataset

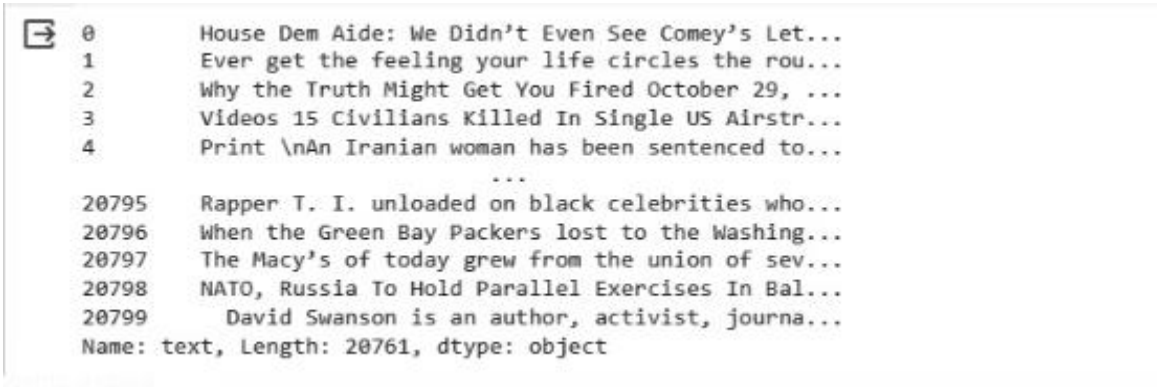


	id	title	author	text
2	20800	Specter of	David Strei	PALO ALTO, Calif. â€" After years of scorning the political process, Silicon Valley has leapt into the fray. The prospect of a President Donald J. Trump is pushing the tech community to move beyond its traditional role a
3	20801	Russian wa	nan	Russian warships ready to strike terrorists near Aleppo 08.11.2016 | Source: Source: Mil.ru Attack aircraft of the Russian aircraft carrier Admiral Kuznetsov get ready to strike terrorists' positions in the vicinity of Aleppo,
4	20802	#NoDAPL:	Common D	Videos
5	20803	Tim Tebov	Daniel Vict	If at first you donâ€™t succeed, try a different sport. Tim Tebow, who was a Heisman quarterback at the University of Florida but was unable to hold an N. F. L. job, is pursuing a career in Major League Baseball. He will
6	20804	Keiser Rep	Truth Broa	42 mins ago 1 Views 0 Comments 0 Likes 'For the first time in history, weâ€™re filming a panoramic video from the station. It means youâ€™ll see everything we see here, with your own eyes. Thatâ€™s to say, youâ€™ll
7	20805	Trump is U	nan	Trump is USA's antique hero. Clinton will be next president 08.11.2016 | Source: AP photo FBI Director James Comey said on November 6 that his department would not be criminally charging Hillary Clinton for revelatic
8	20806	Pelosi Call	Pam Key	Sunday on NBCâ€™s â€œMeet the Press,â€ House Minority Leader Rep. Nancy Pelosi ( ) called for a FBI investigation to find out â€œwhat the Russians haveâ€ on President Donald Trump. Pelosi said, â€œI want to knc
9	20807	Weekly Fe	Trevor Lou	You are
10	20808	Urban Pop	nan	Urban
11	20809		cognitive c	don't we have the receipt?
12	20810	184 U.S. g	Dr. Eowyn	Have you
13	20811	â€œWorki	Doug Diam	Source: CNBC, article by Robert Ferris Arctic sea ice is melting at a rate far faster than anyone thought, and it is already wildly, and perhaps The 36,000 member Institute of Physics â€œClimate geoengineering at scale m
14	20812	The Rise o	Shaun Brad	Written by Shaun Bradley Mandatory vaccinations are about to open up a new frontier for government control. Through the war on drugs, bureaucrats arbitrarily dictate what people can and canâ€™t put into their boc
15	20813	Communis	Steve Wat	Store Communists Terrorize Small Business The owner of the Blue Cat Cafe is the victim of recent terrorist attacks on her business by communists protesters based in Austin Infowars.com - October 27, 2016 Comments
16	20814	Computer	Usa News	
17	20815	Thieves Ta	Melissa Ed	BERLIN â€" You could never palm it, flip it or plunk it into a vending machine. But apparently it can be pinched: One of the worldâ€™s largest gold coins, a Canadian monster called the Big Maple Leaf, was stolen over
18	20816	New Engla	Ken Belson	FOXBOROUGH, Mass. â€" The N. F. L. likes portraying itself as one big family of owners, players and fans who, despite their differences, come together on game days. Yet at the Super Bowl in Houston in two weeks, th
19	20817	College Re	Tom Cicco	The Berkeley College Republicans and the Young Americaâ€™s Foundation have filed a lawsuit against members of the University of California system for their role in restricting an upcoming speaking event featuring An
20	20818	Trump Me	Jason Eash	Trump
21	20819	Visiting Ma	Bryant Rou	If you visit a certain beach in northeastern Madagascar, donâ€™t wear red and donâ€™t even think of speaking French. Across most of the island nation, be very careful where you point, lest your finger accidentally find
22	20820	Reeseâ€™	REALdeal	by ANYA
23	20821	President	REALdeal	President Obama and President-Elect Donald Trump Meet at White House: Share:
24	20822		Dale Johns	VERSE 9.
25	20823	The Real N	Andrew Ar	October
26	20824	Ann Coulte	James Fulf	

test

Fig 6.2 Testing dataset

#Total records



```
0        House Dem Aide: We Didn’t Even See Comey’s Let...
1        Ever get the feeling your life circles the rou...
2        Why the Truth Might Get You Fired October 29, ...
3        Videos 15 Civilians Killed In Single US Airstr...
4        Print \nAn Iranian woman has been sentenced to...
                             ...
20795    Rapper T. I. unloaded on black celebrities who...
20796    When the Green Bay Packers lost to the Washing...
20797    The Macy’s of today grew from the union of sev...
20798    NATO, Russia To Hold Parallel Exercises In Bal...
20799      David Swanson is an author, activist, journa...
Name: text, Length: 20761, dtype: object
```

Fig 6.3 Total records

- **Source code:**

#Importing the necessary libraries

import pandas as pd

import numpy as np

import seaborn as sns

```
import pandas as pd
```

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

#Importing dataset

df = pd.read_csv('/content/drive/MyDrive/csv Fles/train.csv.zip')

```
df =pd.read_csv('train.csv.zip')
```

#Visualizing the dataset

```
df.head(5)
```

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

#Dataset description

```
df.describe()
```

	id	label
count	20800.000000	20800.000000
mean	10399.500000	0.500625
std	6004.587135	0.500012
min	0.000000	0.000000
25%	5199.750000	0.000000
50%	10399.500000	1.000000
75%	15599.250000	1.000000
max	20799.000000	1.000000

#Dataset information

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

#Checking null values

```
df.isnull().sum()
```

```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

#Dropping unnecessary columns

```
df = df.drop(['id', 'title', 'author'], axis = 1)
```

#Visualizing the dataset

```
df.head()
```

	text	label
0	House Dem Aide: We Didn't Even See Comey's Let...	1
1	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired October 29, ...	1
3	Videos 15 Civilians Killed In Single US Airstr...	1
4	Print \nAn Iranian woman has been sentenced to...	1

#Preprocess the text

```
def wordopt(text):
text =text.lower()
text = re.sub('\[.*?\]','', text)
text = re.sub("\\W"," ", text)
text = re.sub('https?://\S+|www\.\S+','', text)
text = re.sub('<.*?>+', '', text)
text = re.sub('[%s]'% re.escape(string.punctuation), '', text)
text = re.sub('\n', '',text)
text = re.sub('\w*\d\w*', '', text)
return text
df['text'] = df['text'].apply(wordopt)
```

```
#Create a function to process the text
def wordopt(text):
  text =text.lower()
  text = re.sub('\[.*?\]','', text)
  text = re.sub("\\W"," ", text)
  text = re.sub('https?://\S+|www\.\S+','', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]'% re.escape(string.punctuation), '', text)
  text = re.sub('\n', '',text)
  text = re.sub('\w*\d\w*', '', text)
  return text

df['text'] = df['text'].apply(wordopt)
```

#Choosing dependent and independent variable

```
x = df['Text']
y= df['label']
```

```
#Dependent and independent variables
x = df['text']
y= df['label']
```

#Splitting the dataset into training and testing

```
#splitting training and testing datas:
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25)
```

After preprocessing, The dataset will be splitted by training data and testing data for predicting this project.

The training data are used to train the model whether the model will be trained perfectly and then the testing data are used to test the pretrained models for its accuracy.

#Converting text into vectors

```
#convert text to vectors
from sklearn.feature_extraction.text import TfidfVectorizer

vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

#Importing Logistic regression

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidVectorizer

from sklearn.linear_model import LogiticRegression

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

#Accuracy for this model prediction

```
pred_lr=LR.predict(xv_test)

LR.score(xv_test, y_test)
```

```
0.9394230769230769
```

#Printing classification reports

```
print(classification_report(y_test, pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      2578
           1       0.94      0.94      0.94      2622

    accuracy                           0.94      5200
   macro avg       0.94      0.94      0.94      5200
weighted avg       0.94      0.94      0.94      5200
```

After testing, it will generate the classification report and print their values.

#Testing the model

```
#Model testing
def output_lable(n):
    if n == 0:
        return "True News"
    elif n == 1:
        return "Fake News"

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    return print("\n\nLR Prediction:{}".format(output_lable(pred_LR[0])))
```

#Prediction

```
news = str(input())
manual_testing(news)
```

```
Donald trump


LR Prediction:Fake News
```

- **Streamlit Framework**

Streamlit is an open-source Python library that simplifies the process of creating web applications. It is designed to be user-friendly, making it accessible to data scientists,

developers, and domain experts who may not have extensive web development experience.

Source code for Creating streamlit environment

```
import streamlit as st
import pandas as pd
import pickle
import re
import string
import json
import requests
from streamlit_lottie import st_lottie
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

st.set_page_config(page_title="Fake News Detector",
page_icon="https://cdn0.iconfinder.com/data/icons/modern-fake-
news/500/asp1430a_1_search_fake_news_icon_outline_vector_thin-512.png",
                   layout='centered',initial_sidebar_state="collapsed")

@st.cache_resource
def bg():
    bg="""
    <style>
    [data-testid="stHeader"]{
        background-color: rgba(0,0,0,0);
    }
    </style>
    """
    st.markdown(bg,unsafe_allow_html=True)
bg()

page_bg_img="""
<style>
[data-testid="stAppViewContainer"]{
background-image:url("https://img.freepik.com/free-photo/concept-fake-
news_23-
```

Fig 6.6 Creating Web app

Creating background image for running my user-friendly streamlit web application

```
st.markdown(page_bg_img,unsafe_allow_html=True)

with st.sidebar:
    st.title("Krithika V")
page_bg_img_side="""
<style>
[data-testid="stSidebar"]{
background-image:url("https://img.freepik.com/free-photo/concept-fake-
news_23-
2148837001.jpg?w=360&t=st=1698133371~exp=1698133971~hmac=b8db5d7c22070c6b
6d0956aaa71d84aef32f59d34a2348d38a4c5feeee30b467");
background-size:fit;
background-color: rgba(0,4,0,9);
</style>
"""
with st.sidebar:
    st.markdown(page_bg_img_side,unsafe_allow_html=True)
```

Fig 6.7 Creating Background image for web app

- **Web application environment**

This is the Login page where authorized users can access the full functionality of the fake news detection system and explore its real-time prediction capabilities

Fig 6.8 Login page

After the user will be login in, it goes to the main page which has the text box and then classify button to classify the user given text and predict the output whether the given news is true or not

Text page:

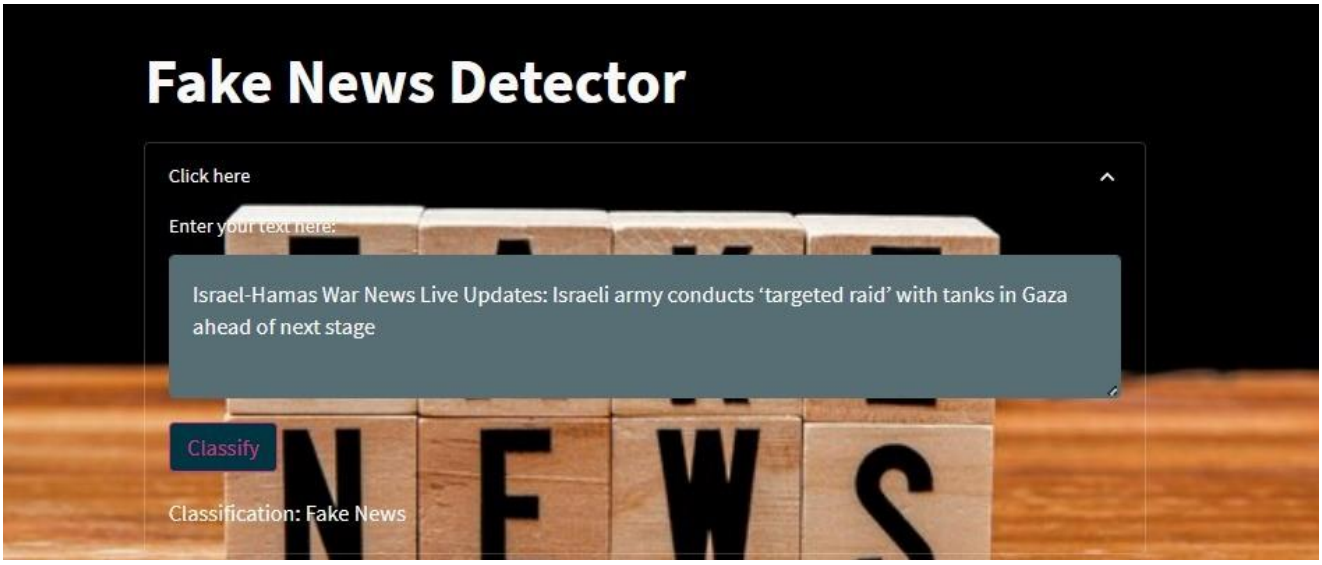


Fig 6.9 Text page

After user entered their text and it can classify whether it is fake or not and then the user goes to exit pages means it exit the text page

Exit page:

Fig 6.10 Exit page

6.1 Model accuracy and Limitations

Explaining model accuracy and its limitations is crucial for providing a comprehensive assessment of the fake news detection project documentation. Users need to understand how well the model performs and the constraints it may have.

- **Accuracy metric**

Accuracy is calculated as the ratio of correct predictions to the total number of predictions. The accuracy for this model prediction is 95%

- **Performance metric**

The confusion matrix shows that how the model was actually predicted correctly by giving the correct text and also predicts the wrong text also.

- **False Negatives and Positives:**

False negatives occur when real news is incorrectly classified as fake, and false positives occur when fake news is incorrectly classified as real. Mention that optimizing for accuracy may lead to trade-offs between these errors.

- **Imbalanced Data:**

When there's a significant class imbalance between real and fake news articles, high accuracy can be achieved by predicting the majority class more frequently.

- **Accuracy Limitations:**

Mention that while high accuracy is desirable, it may not be the sole determinant of the model's effectiveness, especially in fake news detection.

- **User Expectations:**

Users should be aware that the model's accuracy is a reflection of its performance on the dataset it was trained on and that it may not be perfect in all real-world scenarios.

- **Generalization and Overfitting:**

Overfitting, where the model performs exceptionally well on the training data but poorly on new data, can be a limitation that reduces its real-world utility.

- **Changing Data and Trends:**

Highlight that the accuracy of the model may vary as data and news trends change over time. News articles and writing styles evolve, and the model may need updates to maintain its effectiveness.

- **Interpretability:**

While logistic regression is interpretable, some more complex models may achieve higher accuracy. However, this might come at the cost of model interpretability, which is essential for understanding its decisions.

- **Ethical Considerations:**

Model accuracy can be influenced by the training data, and biases in the data can affect predictions. It's important to be transparent about these ethical considerations and limitations.

- **Future Improvements:**

Concluded by discussing how the plan to address these limitations and improve the model in the future. Emphasize the iterative nature of machine learning and the commitment to enhancing the model's accuracy and reliability.

6.2 Contribution to the Fake news Detection

- **Recap the Problem:**

Begin by revisiting the problem that this project aimed to address. Remind the readers of the challenge presented by the spread of fake news and the importance of reliable detection methods.

- **Model Performance:**

Reiterate the model's performance metrics, such as accuracy, precision, recall, F1 score. Emphasize the model's strengths and its effectiveness in classifying news articles.

- **Transparency and Interpretability:**

Stress the importance of model transparency and interpretability in fake news detection. This project shows that how the logistic regression model's coefficients enable users to understand the factors influencing predictions.

- **User-Friendly Interface:**

By using streamlit web application this project shows that user interface for predicting the text by using classification algorithms

- **Real-Time Predictions:**

Reemphasize the value of real-time predictions. This project explains that how this feature provides immediate feedback to users, aiding them in making informed decisions about news article credibility.

- **Transparency and Trust:**

It shows how transparency in the project's design contributes to user trust. Transparency is essential for users to have confidence in the system's decisions.

- **Educational Impact:**

It Explains how it can serve as a valuable educational tool, raising awareness about fake news and the role of machine learning in education.

- **Use in Real-World Scenarios:**

Illustrating the practical application of this project in real-world scenarios. It Explains how individuals, media organizations, and fact-checking initiatives can benefit from the fake news detection system.

- **Ethical Considerations:**

Model accuracy can be influenced by the training data, and biases in the data can affect predictions. It's important to be transparent about these ethical considerations and limitations.

- **Future Improvements:**

Conclude by discussing plan to address these limitations and improve the model in the future. Emphasize the iterative nature of machine learning and the commitment to enhancing the model's accuracy and reliability.

VII. CONCLUSION

Many people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has negative impacts on individual people and society. In this paper, an innovative model for fake news detection using machine learning algorithms has been presented.

This model takes news events as an input and its classify algorithms to predicts the percentage of news being fake or real.

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out.

This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. This study is carried out to check the economic impact that the system will have on the organization

VIII. APPENDIX

8.1 Data extraction

The Fake News Detection dataset was extracted in Kaggle. The dataset used in this project can be accessed at the following link: https://www.kaggle.com/code/therealsampat/fake-news-detection

The dataset screenshot to this project was illustrated below for easy reference:

Fig 8.1 Fake News detection dataset

8.2 Data storage

The Fake News Detection dataset was extracted and stored it in CSV format within local library for easy access and analysis

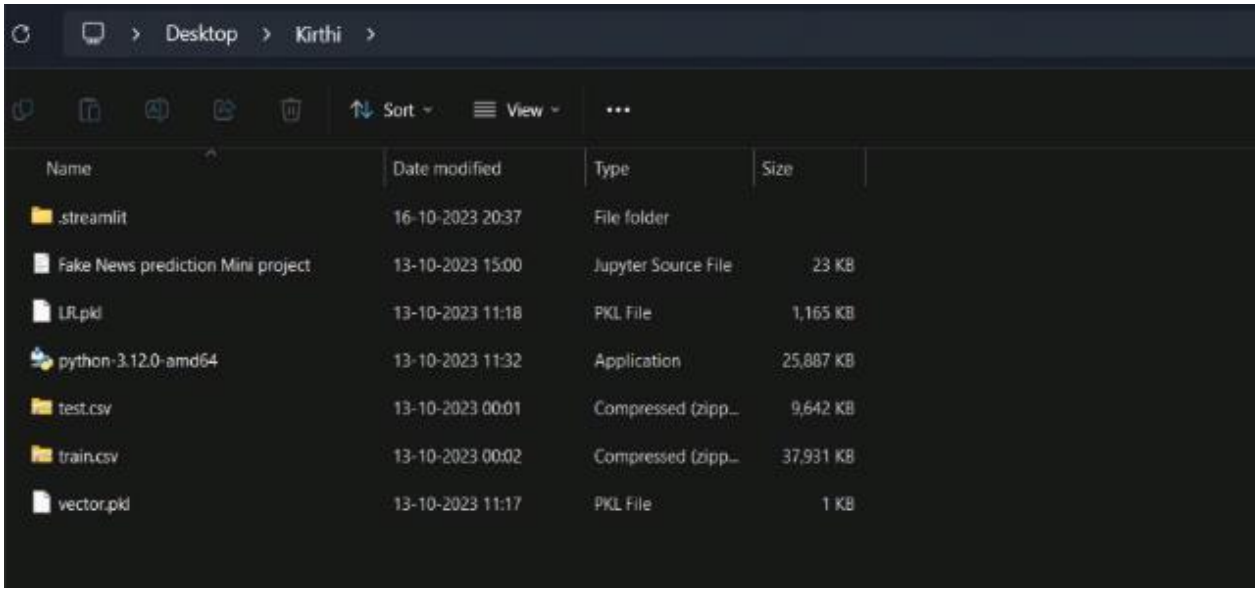
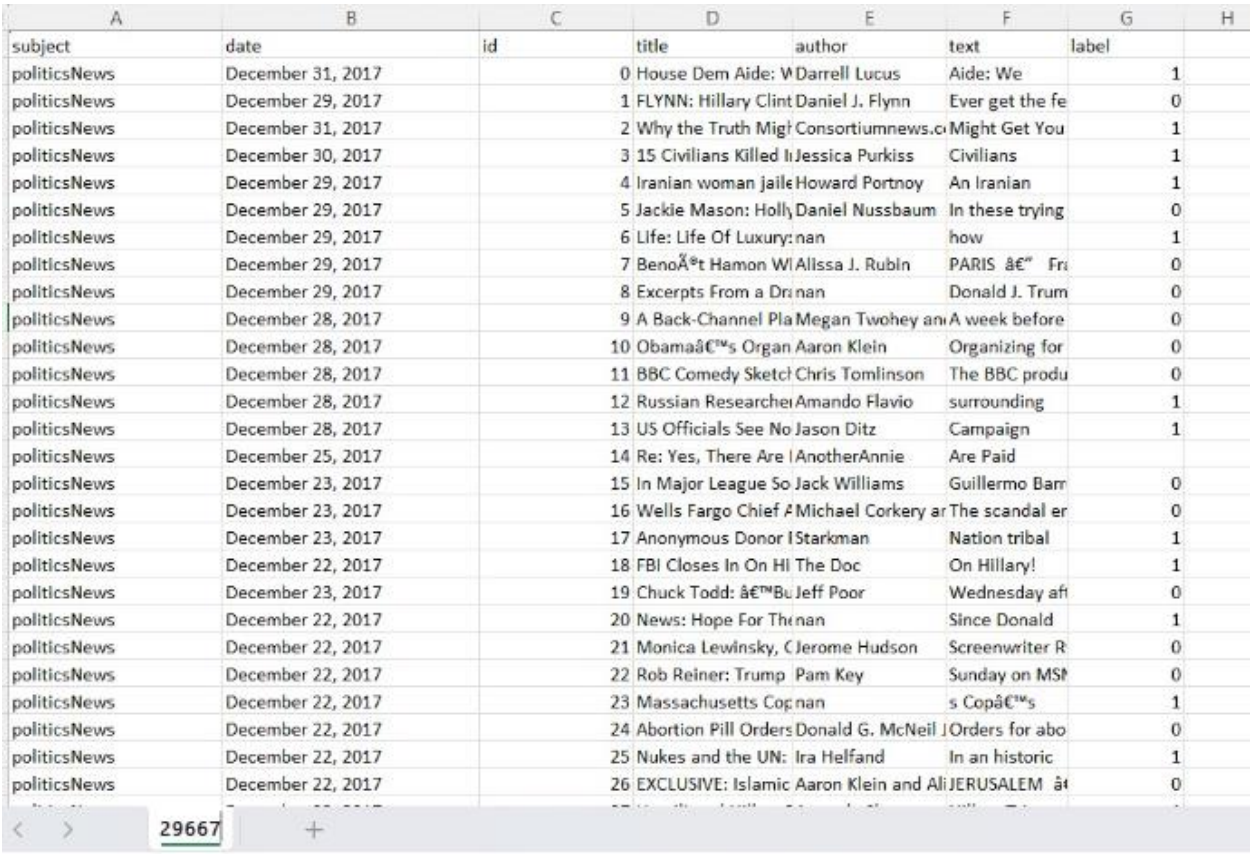


Fig 8.2 Data storage

8.3 Dataset

The Fake News detection dataset was stored in CSV format



	A	B	C	D	E	F	G	H
	subject	date	id	title	author	text	label	
	politicsNews	December 31, 2017	0	House Dem Aide: W	Darrell Lucus	Aide: We	1	
	politicsNews	December 29, 2017	1	FLYNN: Hillary Clint	Daniel J. Flynn	Ever get the fe	0	
	politicsNews	December 31, 2017	2	Why the Truth Migl	Consortiumnews.c	Might Get You	1	
	politicsNews	December 30, 2017	3	15 Civilians Killed I	Jessica Purkiss	Civilians	1	
	politicsNews	December 29, 2017	4	Iranian woman jaile	Howard Portnoy	An Iranian	1	
	politicsNews	December 29, 2017	5	Jackie Mason: Holly	Daniel Nussbaum	In these trying	0	
	politicsNews	December 29, 2017	6	Life: Life Of Luxury:	nan	how	1	
	politicsNews	December 29, 2017	7	BenoÃ®t Hamon Wi	Alissa J. Rubin	PARIS â€" Fr	0	
	politicsNews	December 29, 2017	8	Excerpts From a Dr	nan	Donald J. Trum	0	
	politicsNews	December 28, 2017	9	A Back-Channel Pla	Megan Twohey an	A week before	0	
	politicsNews	December 28, 2017	10	Obamaâ€™s Organ	Aaron Klein	Organizing for	0	
	politicsNews	December 28, 2017	11	BBC Comedy Sketc	Chris Tomlinson	The BBC produ	0	
	politicsNews	December 28, 2017	12	Russian Researcher	Amando Flavio	surrounding	1	
	politicsNews	December 28, 2017	13	US Officials See No	Jason Ditz	Campaign	1	
	politicsNews	December 25, 2017	14	Re: Yes, There Are I	AnotherAnnie	Are Paid		
	politicsNews	December 23, 2017	15	In Major League So	Jack Williams	Guillermo Barr	0	
	politicsNews	December 23, 2017	16	Wells Fargo Chief A	Michael Corkery ar	The scandal er	0	
	politicsNews	December 23, 2017	17	Anonymous Donor I	Starkman	Nation tribal	1	
	politicsNews	December 22, 2017	18	FBI Closes In On HI	The Doc	On Hillary!	1	
	politicsNews	December 23, 2017	19	Chuck Todd: â€™Bu	Jeff Poor	Wednesday aft	0	
	politicsNews	December 22, 2017	20	News: Hope For The	nan	Since Donald	1	
	politicsNews	December 22, 2017	21	Monica Lewinsky, C	Jerome Hudson	Screenwriter R	0	
	politicsNews	December 22, 2017	22	Rob Reiner: Trump	Pam Key	Sunday on MSI	0	
	politicsNews	December 22, 2017	23	Massachusetts Cop	nan	s Copâ€™s	1	
	politicsNews	December 22, 2017	24	Abortion Pill Orders	Donald G. McNeil J	Orders for abo	0	
	politicsNews	December 22, 2017	25	Nukes and the UN:	Ira Helfand	In an historic	1	
	politicsNews	December 22, 2017	26	EXCLUSIVE: Islamic	Aaron Klein and Ali	JERUSALEM â€	0	

29667

Fig 8.3 Dataset as CSV format

8.4 Source code

The Python code is used for implementing logistic regression to classify the news articles in this project.

#Importing necessary libraries

importing pandas as pd

importing numpy as np

importing seaborn as sns

```
import pandas as pd
```

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

#Read the dataset

```
df =pd.read_csv('train.csv.zip')
```

#Visualizing the dataset

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

#Dataset description

	id	label
count	20800.000000	20800.000000
mean	10399.500000	0.500625
std	6004.587135	0.500012
min	0.000000	0.000000
25%	5199.750000	0.000000
50%	10399.500000	1.000000
75%	15599.250000	1.000000
max	20799.000000	1.000000

#Information of the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

#Checking null values

```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

#Handling missing values

```
df=df.fillna('')
```

#Visualizing the dataset

```
id        0
title     0
author    0
text      0
label     0
dtype: int64
```

#Visualizing the columns

```
df.columns
```

```
Index(['id', 'title', 'author', 'text', 'label'], dtype='object')
```

#Dropping the unnecessary columns

```
df = df.drop(['id', 'title', 'author'], axis = 1)
```

#Visualizing the dataset

	text	label
0	House Dem Aide: We Didn't Even See Comey's Let...	1
1	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired October 29, ...	1
3	Videos 15 Civilians Killed In Single US Airstr...	1
4	Print \nAn Iranian woman has been sentenced to...	1

#Importing the necessary metrics

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string
```

#Creating the function to preprocess the text

```
#Create a function to process the text
def wordopt(text):
  text =text.lower()
  text = re.sub('\[.*?\]','', text)
  text = re.sub("\\W"," ", text)
  text = re.sub('https?://\S+|www\.\S+','', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]'% re.escape(string.punctuation), '', text)
  text = re.sub('\n', '',text)
  text = re.sub('\w*\d\w*', '', text)
  return text

df['text'] = df['text'].apply(wordopt)
```

#Dependent and independent variable

```
#Dependent and independent variables
x = df['text']
y= df['label']
```

#Splitting the dataset into training and testing

```
#splitting training and testing datas:
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25)
```

#Converting text into vectors

```
#convert text to vectors
from sklearn.feature_extraction.text import TfidfVectorizer

vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

#Importing vectorizer to preprocess the text

```
vect = TfidfVectorizer()
```

#Importing Logistic regression

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

#Importing pickle to convert web app

```
import pickle
```

#Creating vector file

```
pickle.dump(vect, open('vector.pkl', 'wb'))
```

#Dumping logistic regression

```
pickle.dump(LR, open('LR.pkl', 'wb'))
```

#Loading vector file

```
vector_form=pickle.load(open('vector.pkl', 'rb'))
```

#Loading logistic regression file

```
load_model = pickle.load(open('LR.pkl', 'rb'))
```

#Printing the model accuracy

```
pred_lr=LR.predict(xv_test)

LR.score(xv_test, y_test)
```

#Accuracy value

```
0.9394230769230769
```

#Printing the classification report

```
print(classification_report(y_test, pred_lr))
```

#Classification value

```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      2578
           1       0.94      0.94      0.94      2622

    accuracy                           0.94      5200
   macro avg       0.94      0.94      0.94      5200
weighted avg       0.94      0.94      0.94      5200
```

#If condition

```
#Model testing
def output_lable(n):
    if n == 0:
        return "True News"
    elif n == 1:
        return "Fake News"
```

#Testing the model

```
def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    return print("\n\nLR Prediction:{}".format(output_lable(pred_LR[0])))
```

#User input for model prediction

```
news = str(input())
manual_testing(news)
```

#Model prediction

```
Donald trump


LR Prediction:Fake News
```

8.4 Web development

The Streamlit web development app is to develop this fake news detection project, and then building a model within Streamlit. By inputting a text into the designated text box, the model classifies the news and provides an output indicating whether the given news is fake or not.

Source code:

#Loading the necessary packages

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
import pickle
import re
import string

# Read the dataset
df = pd.read_csv('/content/drive/MyDrive/Csv Files/train.csv.zip')
```

#Data preprocessing

```
# Data preprocessing function
def wordopt(text):
    if isinstance(text, str):
        text = text.lower()
        text = re.sub('\[.*?\]', '', text)
        text = re.sub("\\W", " ", text)
        text = re.sub('https?://\S+|www\.\S+', '', text)
        text = re.sub('<.*?>+', '', text)
        text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
        text = re.sub('\n', '', text)
        text = re.sub('\w*\d\w*', '', text)
    return text

# Remove rows with np.nan in the 'text' column
df = df.dropna(subset=['text'])
```

#Converting text into vectors

```
# Split data into training and testing sets
x = df['text']
y = df['label']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)

# Convert text to vectors
vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

#Saving the vectorizer and model

```
# Save the vectorizer and the model
with open('vector.pkl', 'wb') as vector_file:
    pickle.dump(vectorization, vector_file)

with open('LR.pkl', 'wb') as model_file:
    pickle.dump(LR, model_file)
```

#Model testing

```
# Model Testing
def output_label(n):
    return "True News" if n == 0 else "Fake News"

def manual_testing(news):
    testing_news = {"text": [news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    print(f"LR Prediction: {output_label(pred_LR[0])}")
```

#Generating the confusion matrix

```
# Generate and print the confusion matrix
confusion_mat = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:\n", confusion_mat)
```

```
Classification Report:
              precision    recall  f1-score   support

   True News       0.95      0.94      0.95      2644
   Fake News       0.94      0.95      0.95      2547

    accuracy                           0.95      5191
   macro avg       0.95      0.95      0.95      5191
```

#Confusion matrix result

```
Confusion Matrix:
 [[2497  147]
 [ 128 2419]]
```

8.5 Creating web app

Source code:

#Importing the necessary library

```
import streamlit as st
import pandas as pd
import pickle
import re
import string
import json
import requests
from streamlit_lottie import st_lottie
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
```

#Title for web app

```
st.set_page_config(page_title="Fake News Detector",
page_icon="https://cdn3.iconfinder.com/data/icons/modern-fake-
news/500/asp1430a_1_search_fake_news_icon_outline_vector_thin-512.png",
                   layout='centered',initial_sidebar_state="collapsed")
```

#Background creation

```
@st.cache_resource
def bg():
    bg="""
    <style>
    [data-testid="stHeader"]{
        background-color: rgba(0,0,0,0);
    }
    </style>
    """
    st.markdown(bg,unsafe_allow_html=True)
bg()

page_bg_img="""
<style>
[data-testid="stAppViewContainer"]{
background-image:url("https://img.freepik.com/free-photo/concept-fake-
news_23-
```

#Background image

```
st.markdown(page_bg_img,unsafe_allow_html=True)

with st.sidebar:
    st.title("Krithika V")
page_bg_img_side="""
<style>
[data-testid="stSidebar"]{
background-image:url("https://img.freepik.com/free-photo/concept-fake-
news_23-
2148837001.jpg?w=360&t=st=1698133371~exp=1698133971~hmac=b8db5d7c22070c6b
6dc956aaa71d84aef32f59d34a2348d38a4c5feeee30b467");
background-size:fit;
background-color: rgba(0,4,0,9);
</style>
"""
with st.sidebar:
    st.markdown(page_bg_img_side,unsafe_allow_html=True)
```

#Loading the vectorizer

```
# Load the saved vectorizer and model
with open('vector (1).pkl', 'rb') as vector_file:
    vectorization = pickle.load(vector_file)

with open('LR (1).pkl', 'rb') as model_file:
    LR = pickle.load(model_file)
```

#Data preprocessing function

```
# Data preprocessing function
def wordopt(text):
    if isinstance(text, str):
        text = text.lower()
        text = re.sub('\[.*?\]', '', text)
        text = re.sub("\\W", " ", text)
        text = re.sub('https?://\S+|www\.\S+', '', text)
        text = re.sub('<.*?>+', '', text)
        text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
        text = re.sub('\n', '', text)
        text = re.sub('\w*\d\w*', '', text)
    return text
```

#Function to classify text

```
# Function to classify text
def classify_text(text):
    text = wordopt(text)
    xv_text = vectorization.transform([text])
    prediction = LR.predict(xv_text)
    return "Fake News" if prediction[0] == 1 else "True News"
```

#Streamlit UI

```
# Streamlit UI
st.title("Fake News Detector")

with st.expander('Click here'):
    user_input = st.text_area("Enter your text here:")

    if st.button("Classify"):
        if not user_input:
            st.write("Please enter some text.")
        else:
            result = classify_text(user_input)
            st.write(f"Classification: {result}")
```

#Web page environment

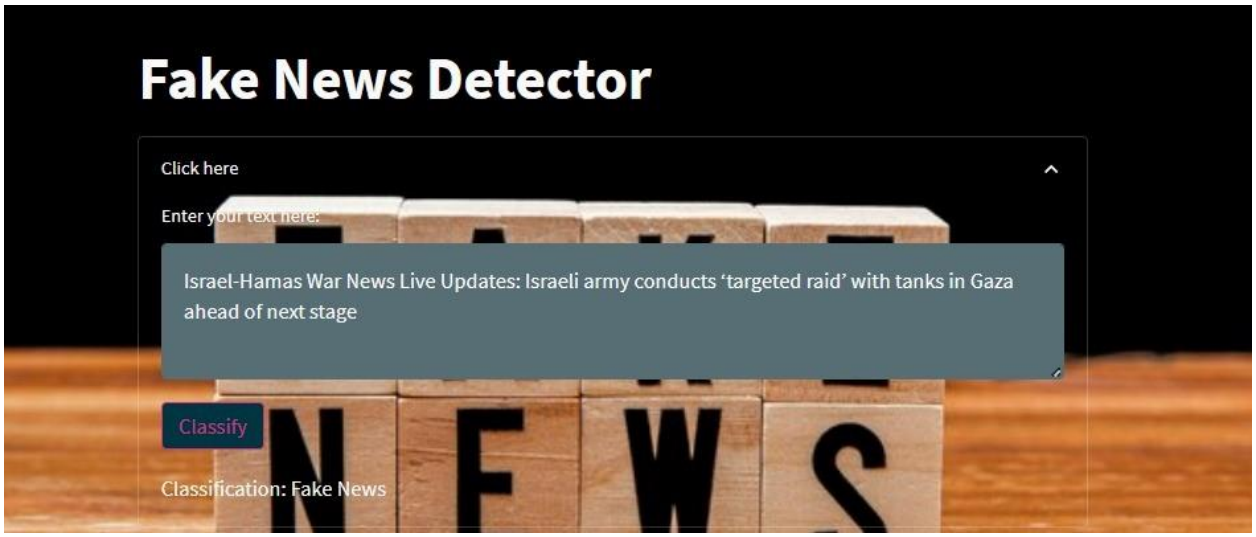


Fig 8.5 Web development environment

IX. FUTURE SCOPE

In the future, plan to enhance this fake news detection project by broadening its capabilities. Currently, this model can predict the authenticity of news articles from a specific dataset. However, aim to take it a step further and make it more versatile. Here are the key areas of future development:

News Source Identification: Work on training the model to recognize the source of the news. This means distinguishing between news from websites, apps, social media, and other platforms. It will allow to provide insights on the credibility of different sources.

Cross-Platform Analysis: The goal is to make this model applicable to a wide range of news platforms. Adapting the model to handle content from websites, news apps, social media posts, and more. This will help users determine the trustworthiness of news across various digital channels.

Real-time Monitoring: Work on a real-time monitoring system that can continuously analyze and report on the authenticity of news as it is published. This will be particularly useful for news agencies and platforms to prevent the spread of fake news.

User-Friendly Interface: Developing a user-friendly interface that simplifies the process of submitting news articles and receiving authenticity ratings will be a key focus. This will make the tool accessible to a broader audience.

X. BIBLOGRAPHY

(i) https://www.simplilearn.com/tutorials/machine-learning-tutorial/how-to-create-a-fake-news-detection-system

(ii) https://www.analyticsvidhya.com/blog/2022/08/step-by-step-explanation-of-text-classification/

(iii) https://www.geeksforgeeks.org/understanding-logistic-regression/

(iv) https://blog.streamlit.io/how-to-build-streamlit-apps-on-replit/