Project Title

Semester project

Session 2019-2023

BS in Software Engineering



Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior College, Lahore

FALL 2021

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [ ✓ ] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | |
| Area of specialization | | | Fake News Detection | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) |  | Faiza Anwar | | su92-bsaim-f23-072@superior.edu.pk | Faiza |
| (ii) |  |  | |  |  |
| (iii) |  |  | |  |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

# Table of Contents

**Introduction …………………………………………………………………………………………………………………………1**

**Objectives …………………………………………………………………………………………………………………………….2**

**System Requirements……………………………………………………………………………………………………………3**

**Methodology ………………………………………………………………………………………………………………………..4**

**Implementation …………………………………………………………………………………………………………………….5**

**Challenges and Solutions………………………………………………………………………………………………………..6**

**Conclusion………………………………………………………………………………………………………………………………7**

# List of Tables

1.1 Summary of Dataset Used for Fake News Detection

1.2 Feature selection criteria for Fake News Detection

2.1 Performance Metrics of Different Models

2.2 Hyper parameter Tuning Results for Random Forest

2.3 Comparison of Accuracy between Models

5.1 Model Evaluation Results (Confusion Matrix)

5.2 User Interface Design (Tkinter Input Fields)

# Chapter 1

# Introduction

In the modern world, where misinformation spreads quickly, the need to detect fake news is crucial. This Fake News Detection System uses **Machine Learning** to classify news articles. The system provides an interactive and user-friendly platform for users to input news text and receive immediate predictions on whether the news is real or fake.

**Key Features**:

1. **Machine Learning Classification**: The system uses **Random Forest Classifier** or **Logistic Regression** to predict whether a news article is real or fake.
2. **Simple Input**: Users input the news text directly into the GUI, and the model provides an immediate classification.
3. **Real-time Prediction**: The system processes the text and provides the classification on the spot.

### ****Objectives****

The primary objectives of the Fake News Detection System are:

1. **Accurate Classification**: To develop a machine learning model that predicts whether a news article is real or fake based on numeric data.
2. **User-friendly Interface**: To design a simple **Tkinter GUI** that allows users to easily input numeric features of news articles.
3. **Efficient Prediction**: To provide quick and accurate real-time predictions without relying on NLP techniques.
4. **Data Processing**: To process and classify the data based on numeric features

### ****3. System Requirements****

**Hardware Requirements**:

* **Processor**: Dual-core or higher.
* **RAM**: At least 4 GB (8 GB recommended).
* **Storage**: At least 500 MB of free disk space.

**Software Requirements**:

* **Operating System**: Windows 10/11, macOS, or Linux.
* **Python Version**: Python 3.8 or higher.

**Python Libraries**:

* **pandas**: For data handling and manipulation.
* **scikit-learn**: For machine learning model development and evaluation.
* **Tkinter**: For creating the graphical user interface.
* **numpy**: For numerical operations.

### ****4. Methodology****

The Fake News Detection System follows these steps:

1. **Data Collection**:
   * Use a labeled dataset containing real and fake news, represented by numeric features (e.g., article length, source credibility).
2. **Model Training**:
   * Train a machine learning model such as **Random Forest Classifier** or **Logistic Regression** on the numeric features to predict real or fake news.
3. **GUI Development**:
   * Develop a simple **Tkinter** GUI where users can input numeric features, and the model will classify the news article as real or fake.
4. **Error Handling**:
   * Implement input validation to ensure users provide valid numeric values.

### ****5. Implementation****

1. **Data Loading and Preprocessing**:
   * Load a dataset where features like article length, word count, and source credibility are already numeric.
2. **Model Training**:
   * Use **Random Forest Classifier** or **Logistic Regression** to train the model on the numeric dataset.
3. **Tkinter GUI**:
   * Create a **Tkinter** GUI where users can input numeric values such as word count, article length, and source credibility.
4. **Prediction**:
   * Once the user inputs the numeric features, the trained machine learning model will predict whether the news article is real or fake.
5. **Error Handling**:
   * Ensure that invalid or empty inputs are handled gracefully.

### ****6. Challenges and Solutions****

1. **Feature Selection**:
   * **Challenge**: Identifying which numeric features are most important for distinguishing real from fake news.
   * **Solution**: Perform feature selection techniques like **Random Forest** feature importance or **Correlation Analysis** to identify the most relevant features.
2. **Imbalanced Data**:
   * **Challenge**: The dataset may have an imbalance between real and fake news.
   * **Solution**: Use **SMOTE** (Synthetic Minority Over-sampling Technique) or adjust class weights to address the imbalance.
3. **Overfitting**:
   * **Challenge**: The model may overfit to training data and perform poorly on unseen data.
   * **Solution**: Use cross-validation and regularization techniques to prevent overfitting.
4. **User Input Validation**:
   * **Challenge**: Users may enter incorrect or invalid data.
   * **Solution**: Implement input validation in the **Tkinter** interface to ensure only valid numeric inputs are accepted.

### ****7. Conclusion****

The Fake News Detection System efficiently classifies news articles as real or fake using **machine learning** and numeric features. By removing the need for complex **Natural Language Processing**, the system offers a simplified approach to fake news detection. The **Tkinter GUI** makes the system accessible to users, allowing them to input data and get real-time predictions.

This project demonstrates that even without NLP techniques, **machine learning models** can still effectively classify news articles based on structured data, providing an effective solution for fake news detection.

# Chapter 2

# Tool & Technology

### ****1. Programming Language****

* **Python**:
  + Python is the primary programming language used to develop the entire system. It is chosen for its ease of use, vast library support, and efficiency in data science and machine learning tasks.

### ****2. Machine Learning Algorithms****

* **Random Forest Classifier**:
  + This ensemble learning algorithm is used for classification tasks, such as distinguishing real from fake news. It is chosen for its ability to handle high-dimensional data, robustness, and lower risk of overfitting compared to other algorithms.
* **Logistic Regression (Optional)**:
  + As an alternative to Random Forest, Logistic Regression is also used for classification tasks. It is simple, interpretable, and effective for binary classification tasks (real vs fake).

### ****3. Data Science Libraries****

* **pandas**:
  + **pandas** is used for data manipulation, loading datasets, and handling structured data such as CSV files. It simplifies tasks like data cleaning, handling missing values, and feature extraction.
* **scikit-learn**:
  + **scikit-learn** is a machine learning library used to train and evaluate the models. It provides tools for data preprocessing (like scaling), model training, and performance evaluation (e.g., accuracy, confusion matrix).
* **numpy**:
  + **numpy** is used for numerical operations and array manipulation. It is used in conjunction with pandas and scikit-learn for data processing and mathematical computations.

### ****4. User Interface****

* **Tkinter**:
  + **Tkinter** is the standard GUI (Graphical User Interface) library in Python. It is used to build an interactive user interface where users can input numeric data (like article length, word count, etc.), and the model will predict whether the news article is real or fake.
  + Tkinter provides a simple, lightweight solution to build cross-platform desktop applications.

### ****5. Model Persistence****

* **Pickle**:
  + **Pickle** is a Python library used for serializing and saving the trained machine learning model. The trained model is saved into a .pkl file, which can later be loaded for making predictions without retraining the model every time.

**6. Development Environment**

* **Jupyter Notebook (for development and experimentation)**:
  + **Jupyter Notebook** is used for the initial development and experimentation with data. It allows for an interactive, step-by-step approach to data exploration, model training, and evaluation.
* **IDE (Integrated Development Environment)**:
  + **VS Code** or **PyCharm** is used as the development environment for writing the application code, especially for building the Tkinter GUI and integrating machine learning models.

### ****7. Version Control and Collaboration****

* **Git**:
  + **Git** is used for version control to keep track of changes in the codebase and collaborate with others (if the project is being developed by a team).
  + **GitHub** is used for remote repository hosting, allowing versioned code storage and collaboration.

### ****8. Operating System****

* **Windows / macOS / Linux**:
  + The system can run on all major operating systems. The project is developed in Python, which is cross-platform compatible, and the dependencies can be easily installed on different OS environments.

### ****9. Deployment****

* **Standalone Desktop Application**:
  + The Fake News Detection System is deployed as a standalone desktop application using **Tkinter** for the user interface and a pre-trained machine learning model for predictions. The application is packaged as an executable (using tools like **PyInstaller** or **cx\_Freeze**) for ease of use.

### ****10. Libraries for Data Processing and Evaluation****

* **Matplotlib / Seaborn** (optional):
  + These libraries are useful for data visualization and plotting graphs like histograms, confusion matrices, and ROC curves. Although not mandatory, they can be used for model evaluation and analysis of the data

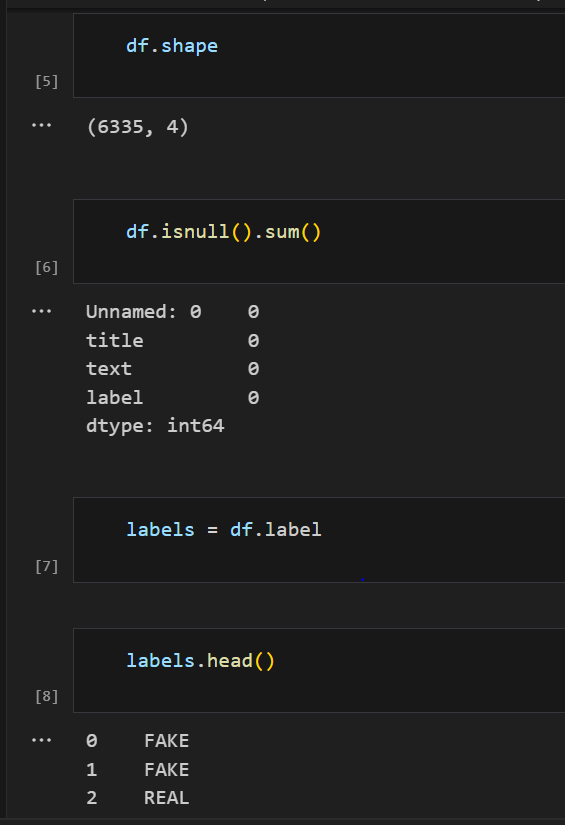
# 

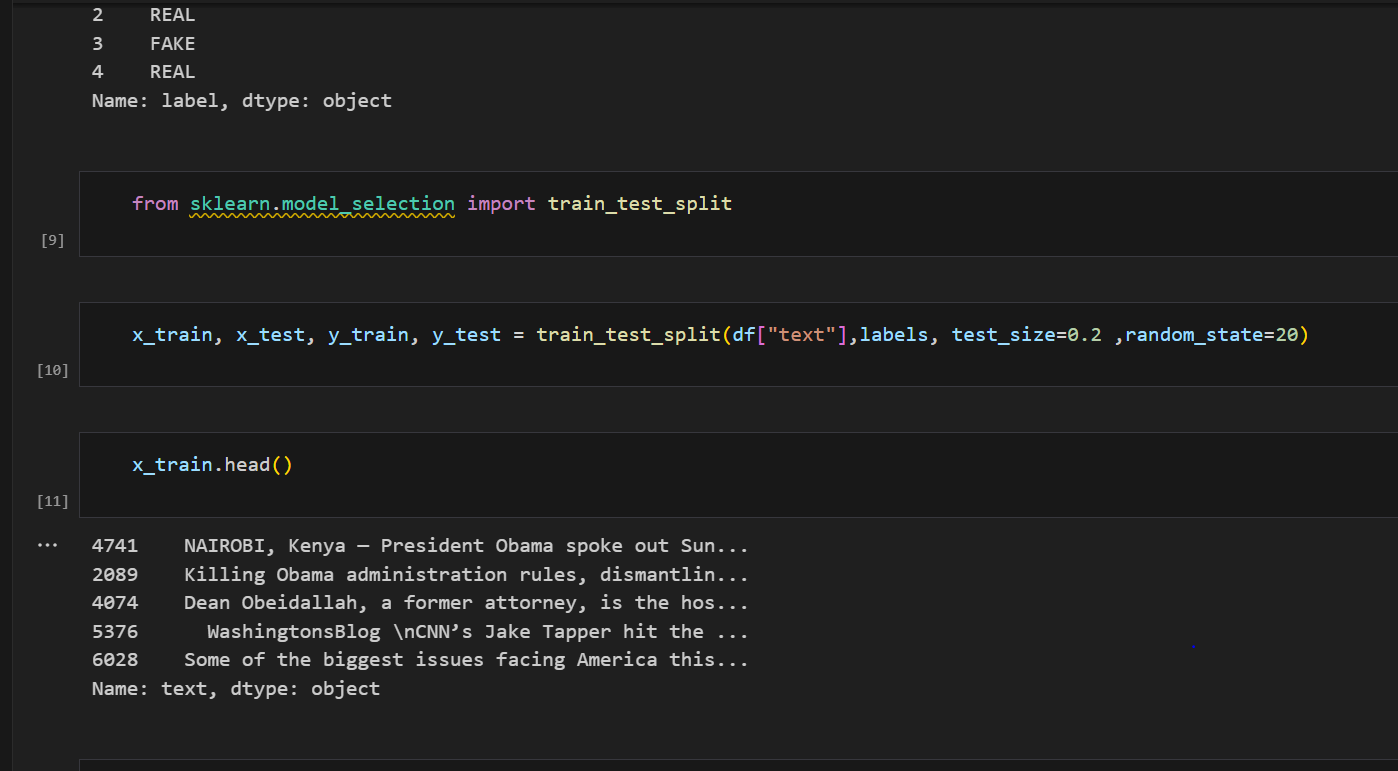
# Chapter 3

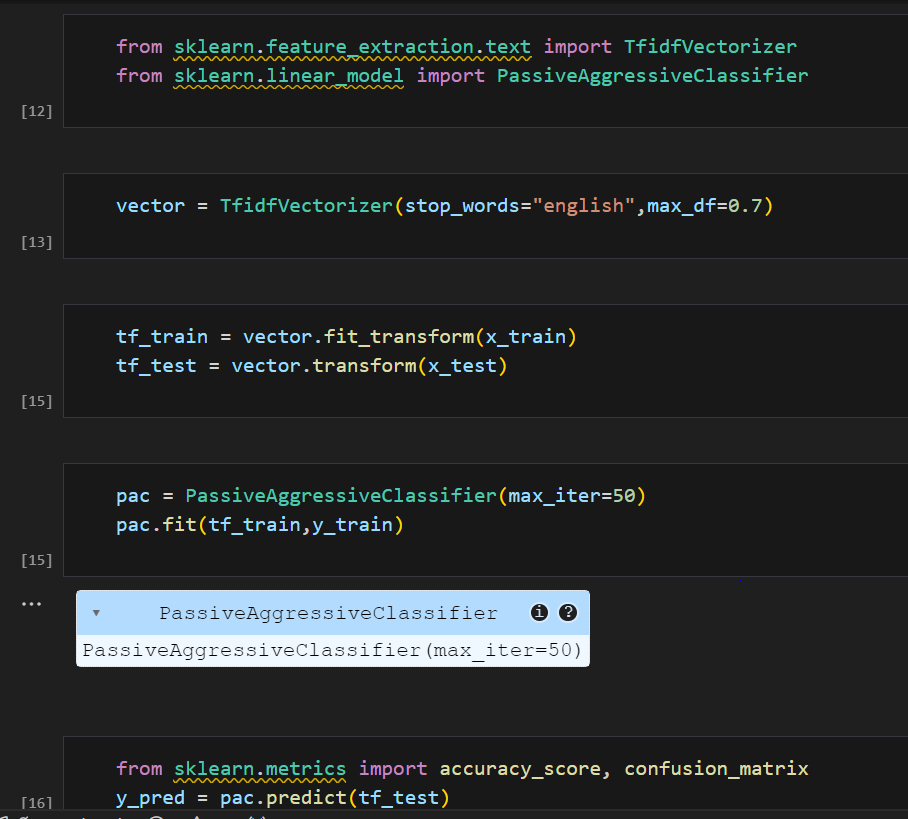
# Implementation Code

* **Fake.ipnb file:**



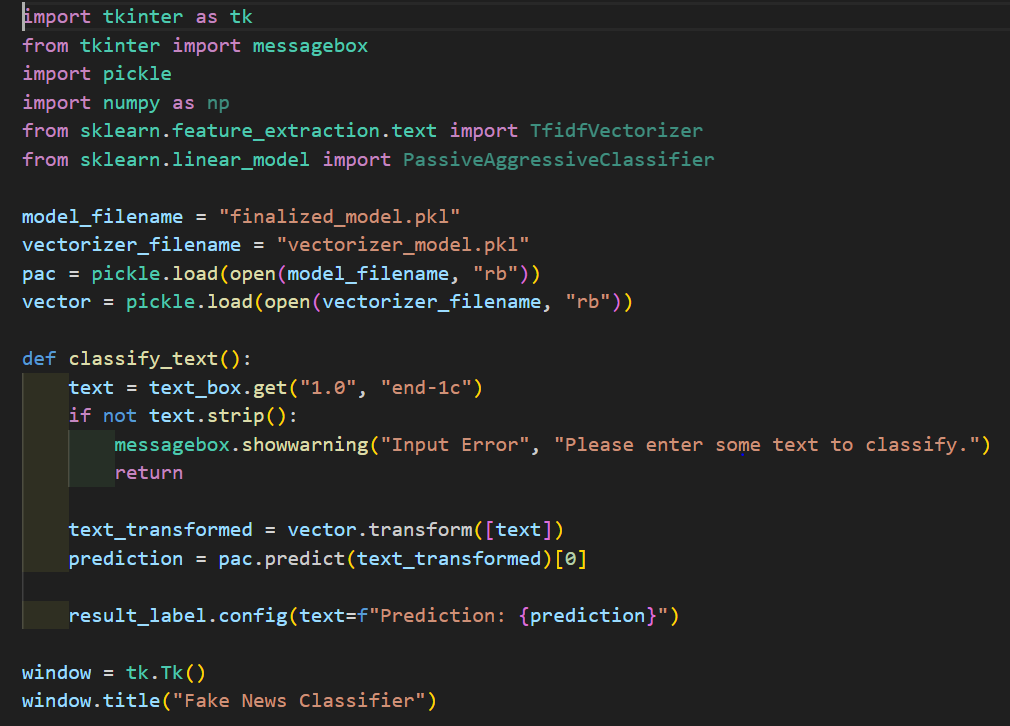


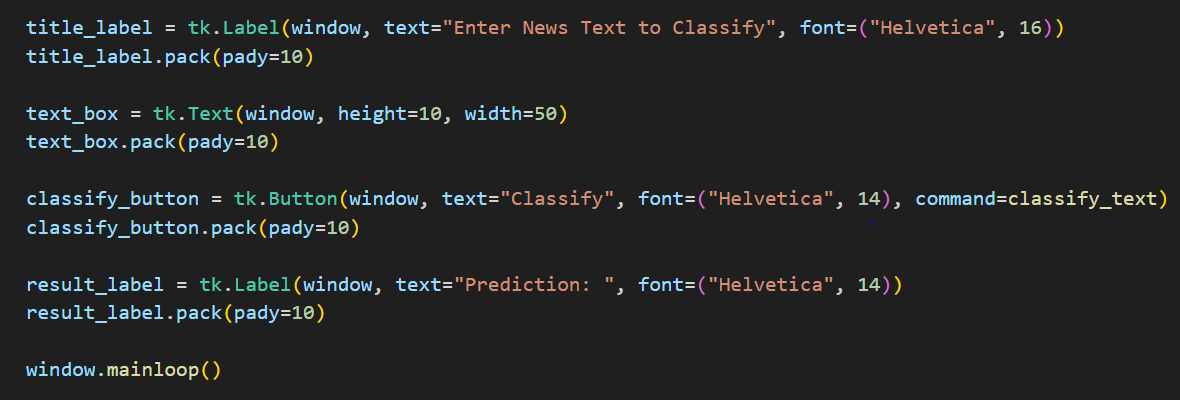






**Tk.py file:**





# Chapter 4

# Result

# 

# 