Fake news detection

# MUHAMMAD SHAHBAZ

# MUHAMMAD USAMA SHARIF



DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD, WAH CAMPUS

WAH CANTT – PAKISTAN

SESSION 2021-2025

Fake news detection

*Undertaken By:*

# MUHAMMAD SHAHBAZ

REG. NO. CIIT/SP21-BCS-049/WAH

# MUHAMMAD USAMA SHARIF

REG. NO. CIIT/SP21-BCS-043/WAH

# 

# *Supervised By:*

# Dr. SAEED UR REHMAN



A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF BACHELORS IN COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD, WAH CAMPUS

WAH CANTT – PAKISTAN

SESSION 2021-2025

**DEDICATION**

There are a lot of people to whom we should give dedication to the completeness of our final year project. Firstly, we would like to thank Allah Almighty the most Merciful and most Beneficent, without the support of the Lord we are nothing. After that we dedicated this project to our beloved parents as without their support, we wouldn’t be able to stand here today. We also dedicate this project to our teachers whose guidance and motivation helped us to cover this milestone. We would also like to thank our few colleagues and seniors for helping with our project. All in all, the credit of this project really goes to our loving parents and our supervisor because both (parents and supervisor) not only guided us but built courage, provided motivation and always believed in us.

**ACKNOWLEDGEMENT**

Once again, we would firstly like to thank Allah Almighty the most merciful and most beneficent as we are nothing our skills are nothing if we don’t have Allah Almighty’s support in life. Next Comes our parents the people who are the guardians, the mentor, in short words who are everything because throughout the life they always stood beside us helped us supported us, has fulfilled our requirements, needs, necessities, wishes everything and still here for us. Afterwards, a special thanks to our supervisor SIR SAEED UR REHMAN, Tenured Associate Professor at COMSATS University Islamabad, Wah Campus who always helped us in case of project and in other issues as well. We are lucky to have him as our teacher, our mentor, a guideline for our success. Due to his guidance and supervision this project has been completed, so special thanks to our supervisor for being a supervisor for the help, for the support, the guidance for everything.

**PROJECT BRIEF**

|  |  |
| --- | --- |
| **PROJECT NAME:** | **Fake News Detection using NLP** |
| **PROJECT**  **OBJECTIVE:** | **Fake News Detection using NLP** is a process that applies **Natural Language Processing (NLP)** techniques to identify whether a piece of news or information is false or misleading. This is important because misinformation can spread rapidly, and automatic detection can help mitigate its impact |
| **UNDERTAKE**  **N BY:** | MUHAMMAD SHAHBAZ  MUHAMMAD USAMA SHARIF |
| **SUPERVISED BY:** | DR. SAEED UR REHMAN  DEPARTMENT OF COMPUTER    SCIENCES CUI, WAH CAMPUS  PAKISTAN |

**ABSTRACT**

**Fake News Detection using NLP** is a process that applies **Natural Language Processing (NLP)** techniques to identify whether a piece of news or information is false or misleading. This is important because misinformation can spread rapidly, and automatic detection can help mitigate its impact

### **Frontend: ReactJS**

#### File: App.js

**Key Features:**

1. **User Interface:**
   * A simple form for users to input news articles for analysis.
   * Displays the result ("Real" or "Fake") returned from the backend.
2. **State Management:**
   * useState is used to manage the user input (news), result (result), and loader state (loading).
3. **Backend Communication:**
   * Sends the user input to the backend API (/predict) using the fetch method.
   * Displays the backend's prediction result or an error message.
4. **Error Handling:**
   * Handles network or server errors gracefully.
   * Displays an appropriate error message when a connection fails.
5. **UI Elements:**
   * Includes a loader animation during the request.
   * Provides a responsive and accessible input form.

### **Backend: Flask**

#### File: app.py

**Key Features:**

1. **Flask Setup:**
   * The Flask application is initialized with CORS support to enable communication with the ReactJS frontend.
2. **Model Integration:**
   * A pre-trained machine learning model and vectorizer are loaded using joblib.load.
   * These are stored in a .pkl file (fake\_news\_detection.pkl).
3. **API Endpoints:**
   * **/:** A basic route to verify the server's functionality.
   * **/predict:** Accepts POST requests with news text as input and returns predictions.
4. **Text Preprocessing:**
   * Utilizes a vectorizer to transform the input text into a format suitable for the model.
5. **Prediction Logic:**
   * Maps model output (0 or 1) to human-readable labels ("Fake" or "Real").
   * Includes validation for empty inputs and detailed logging for debugging.
6. **Error Handling:**
   * Handles exceptions during prediction and logs errors.
   * Returns appropriate error messages to the frontend.

### **Machine Learning Model**

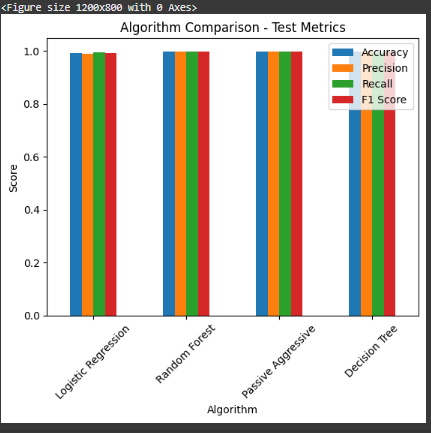
#### File: fake\_news\_detection.pkl

* A serialized file containing:
  1. A machine learning model (likely trained on labeled fake/real news data).
  2. A vectorizer for transforming text into a numeric representation (e.g., TF-IDF or Count Vectorizer).

**Key Assumptions:**

* The model predicts binary outcomes:
  + 0 for Fake
  + 1 for Real
* The vectorizer ensures compatibility between the input format and the model.

### **Flow**

1. **User Input:**
   * The user enters a news article in the ReactJS app and submits it.
2. **Backend Communication:**
   * The frontend sends a POST request to the /predict API.
3. **Prediction:**
   * The backend preprocesses the input, passes it to the model, and maps the output to a label.
4. **Response:**
   * The prediction ("Fake" or "Real") is sent back to the frontend.
5. **Output:**
   * The result is displayed to the user in the ReactJS app.
   * 

### **Recommendations**

1. **Security:**
   * Sanitize user inputs to avoid potential vulnerabilities like SQL injection or XSS.
   * Use HTTPS for secure communication.
2. **Scalability:**
   * Deploy the app using a cloud service (e.g., AWS, Heroku) for better scalability.
   * Add caching for frequently accessed data.
3. **User Experience:**
   * Include more detailed feedback, such as confidence scores or explanations for predictions.
   * Add a history feature to let users view past analyses.
4. **Performance:**
   * Optimize the vectorizer and model for quicker responses, especially if dealing with high traffic.
5. **Logging and Monitoring:**
   * Use a centralized logging solution like ELK Stack or AWS CloudWatch for enhanced monitoring.

**Output screen:**

****