Software Requirements Specification (SRS)

Fake News Detection Web Application

1. Introduction

Purpose:

This document specifies the requirements for a Fake News Detection Web Application

that classifies news as Real or Fake using Logistic Regression, LSTM, and BERT models.

Scope:

The system allows users to input news text, select a model, and receive prediction results

with confidence scores. The system is built using Python Flask and deployed on Render.

2. Overall Description

System Components:

- Frontend (HTML/CSS)

- Backend (Flask)

- Machine Learning Models (Logistic Regression, LSTM, BERT)

- Deployment on Render Cloud

Architecture Flow:

User → Web UI → Flask Server → Model → Prediction → Response

3. Functional Requirements

- User can input news text.

- User can select model (Logistic Regression / LSTM / BERT).

- System preprocesses text.

- System predicts Fake or Real.

- System displays confidence score and model name.

Model Requirements:

Logistic Regression:

- TF-IDF Vectorization

- Pre-trained .pkl model

LSTM:

- Tokenization and Padding

- Pre-trained .h5 model

BERT:

- HuggingFace Transformers

- Softmax probability output

4. Non-Functional Requirements

Performance:

- Response time under 5 seconds.

- Supports concurrent users.

Security:

- Input validation
- HTTPS deployment

Usability:

- Simple and responsive interface

5. Technology Stack

- Python 3
- Flask
- Scikit-learn
- TensorFlow / PyTorch
- Transformers
- Gunicorn
- Render Deployment

6. API Endpoint

POST /predict

Input JSON:

```
{
"text": "News content",
"model": "bert"
}
```

Output JSON:

```
{
"prediction": "Fake",
"confidence": 0.92
}
```

7. Deployment Steps

- Push project to GitHub.
- Connect repository to Render.
- Set Build Command: pip install -r requirements.txt
- Set Start Command: gunicorn app:app
- Deploy Web Service.

8. Constraints

- Large BERT model size.
- Limited RAM in free deployment tier.
- No GPU support in free plan.

9. Future Enhancements

- Add explainable AI (LIME/SHAP)
- Add multilingual support
- Add user authentication
- Add prediction history database

Conclusion:

The system provides scalable fake news detection using ML and DL models,

deployed efficiently using Flask and Render.