Project Proposal

Project Name:AI-Powered Fake News Detection and Network Analysis

PROJECT DESCRIPTION (< 150 words)
This project aims to develop a comprehensive AI-powered system for detecting and analyzing fake news. It combines advanced Natural Language Processing (NLP) models (such as BERT or LSTM) to classify news articles as real or fake, with a graph-based approach (using NetworkX) to model and visualize how misinformation propagates through social networks. Interactive dashboards built with Bokeh will highlight key misinformation hubs and trends, enabling users—including journalists and fact-checkers—to quickly identify unreliable sources. By tracking suspicious articles and mapping their sharing patterns, the system provides actionable insights to help curb the spread of fake news on social media platforms.

SOLUTION (Deliverables). Write a bullet point list of what you expect your software will achieve. I do not hold you to this list for your end-product.

- **Fake News Classifier**: A deep learning model (BERT/LSTM) to label articles as real or fake.
- **Network Graph Construction**: Building a social graph (NetworkX) to trace how misleading content spreads among users.
- **Interactive Visualizations**: Bokeh-based dashboards showing misinformation hubs, trends, and propagation paths.
- **Reporting Tools**: Summaries and alerts to highlight emerging fake news stories and their

DATASETS (if any used).

- Possible publicly available fake news datasets (e.g., FakeNewsNet, LIAR dataset, or Kaggle Fake News dataset).
- Social media or user-sharing datasets (synthetic or anonymized real-world data) for network analysis.

Expected Tools (Cloud DBs, Hardware, & Python Libraries to be used.

- **Cloud Databases**: AWS RDS or MongoDB Atlas (for storing news data and user interactions).
- **Hardware**: Local machine with a GPU or cloud-based GPU instances (e.g., AWS EC2) for model training.
- Python Libraries:
  - *Transformers/HuggingFace* for BERT implementation
  - *TensorFlow* or *PyTorch* for deep learning
  - *NetworkX* for graph-based analysis
  - *Bokeh* (or Plotly) for interactive data visualization
  - *scikit-learn* for additional machine learning tasks

Rough Timeline (Fill in the columns):

Weeks Project Task Timeline

week	Plan
1	- **Project Setup & Data Gathering** • Finalize project scope and objectives. • Set up environment (Python, required libraries, cloud resources). • Identify and gather fake news datasets and any additional data.
2	- **Data Preprocessing & Baseline Model** • Clean and preprocess text data (handle duplicates, missing values). • Conduct Exploratory Data Analysis (EDA) to understand distributions and text characteristics. • Implement a baseline NLP model (e.g., simple LSTM or basic BERT).
3	- **Model Training & Tuning** • Train the chosen NLP model on prepared data. • Perform hyperparameter tuning to improve performance. • Evaluate metrics (accuracy, precision, recall, F1) and refine model.
4	- **Network Construction & Analysis** • Use NetworkX to build a social graph (users, articles, sharing relationships). • Identify network properties (centrality, clustering) to locate misinformation hubs. • Begin implementing interactive visualizations (Bokeh).
5	- **Integration & Visualization** • Integrate classification results with the network analysis pipeline. • Develop and refine Bokeh dashboards to illustrate fake news propagation trends. • Start system testing and gather feedback.
6	- **Testing, Documentation & Final Presentation** • Conduct thorough end-to-end testing of the system. • Prepare comprehensive documentation (user guide, technical report). • Finalize and present the project, highlighting key results and future work.