# Financial News Sentiment Analyzer

**1. Overview**

This project focuses on developing an NLP-based system that analyzes and classifies the sentiment of financial news headlines. The system will categorize news as **positive**, **negative**, or **neutral**, aiding users in making informed decisions based on market sentiment.

**2. Objectives**

* Build an effective NLP pipeline to preprocess and vectorize financial news headlines.
* Train and optimize a sentiment classification model using financial domain-specific datasets.
* Deploy an accessible web-based application for real-time sentiment analysis of financial news.

**3. Functional Requirements**

* **Text Preprocessing**:
  + Clean and tokenize news headlines.
  + Implement NLP techniques such as TF-IDF, Word2Vec embeddings, or transformer-based embeddings (e.g., BERT, FinBERT).
* **Model Training**:
  + Train classification algorithms including:
    - Traditional models like Logistic Regression
    - Deep learning models such as LSTM (Long Short-Term Memory networks)
    - Specialized transformer models like FinBERT fine-tuned on financial text.
* **Sentiment Classification**:
  + Output sentiment labels: **Positive**, **Negative**, **Neutral**.
* **Deployment**:
  + Serve the model using frameworks such as Flask or FastAPI, or build an interactive UI using Streamlit for users to input news headlines and get instant sentiment results.

**4. Implementation Strategy**

**Phase 1: Data Preparation**

* Collect and clean Financial PhraseBank dataset (remove noise, stopwords, punctuation).
* Tokenize headlines and prepare text vectors using:
  + **TF-IDF** for term relevance weighting.
  + **Word2Vec** for word embeddings capturing contextual similarity.
  + **Transformer embeddings** (FinBERT) for deep contextual understanding.

**Phase 2: Model Development**

* Train baseline models like Logistic Regression on TF-IDF vectors.
* Develop LSTM networks to capture sequential dependencies in text.
* Fine-tune FinBERT on the financial dataset for improved semantic understanding.
* Perform hyperparameter tuning to optimize accuracy, precision, and recall.

**Phase 3: Deployment**

* Create backend API to accept news headlines and return sentiment prediction.
* Develop frontend with Streamlit or Flask UI for user-friendly interactions.
* Ensure real-time analysis with minimal latency for practical usability.

**5. Expected Outcome**

* A robust sentiment analysis system that accurately classifies financial news headlines.
* Tool to help traders and investors gauge market mood and make informed decisions.
* Scalable and extendable platform for incorporating additional data sources or sentiment categories in the future.