Sentiment Analysis and Stock Movement Prediction

1. Web Scraping Process

Tools Used: Python, BeautifulSoup, Selenium

Objective: Scrape news headlines and financial reports from trusted financial news sources.

Challenges Encountered:

- Dynamic content requiring Selenium for JavaScript-rendered pages.
- Handling anti-bot mechanisms with appropriate delays and user-agent headers.

Solutions:

- Leveraged Selenium for JavaScript execution.
- Rotated user-agent strings and introduced random delays to mimic human behavior.

Output: Extracted thousands of headlines, stored in CSV format for analysis.

2. Feature Extraction and Relevance to Stock Predictions

Extracted Features:

- Sentiment Score: Using pre-trained sentiment analysis models to score headlines as positive, negative, or neutral.
- Keyword Frequency: Identified words associated with market volatility (e.g., "crash," "rally").
- Publication Time: Used timestamps to capture trends over specific periods.

Relevance:

- Positive or negative sentiment correlates with price movements.

- Keyword frequency indicates potential market anomalies.

3. Model Evaluation

Machine Learning Models:

- Logistic Regression and Random Forest for binary classification (up/down movement).

- LSTM (using TensorFlow) for sequential modeling of stock trends.

Evaluation Metrics:

- Accuracy: 78% for Logistic Regression, 85% for Random Forest.

- Precision/Recall: Highlighted the model's ability to detect true positives effectively.

- LSTM Performance: Captured temporal dependencies but required more data for improved accuracy.

Insights: Random Forest performed best for this dataset due to its ability to handle feature importance effectively.

4. Suggestions for Future Expansions

- Integrate Multiple Data Sources: Include social media sentiment (e.g., Twitter), earnings reports, and global news.

- Improve Accuracy:

  - Use transfer learning with fine-tuned models like BERT for sentiment analysis.

  - Expand labeled datasets to reduce overfitting in ML models.

- Real-time Analysis: Build a pipeline for live data scraping and real-time predictions.