Stock Movement Prediction Using Reddit Sentiment Analysis and LSTM

NAME: YASH AGARWAL

GITHUB LINK: https://github.com/yashh-jiit/ML-Assn

1. Introduction

This project aims to predict stock price movements by combining sentiment analysis of Reddit posts with historical stock data. Sentiment scores derived from Reddit posts are integrated with stock features to train a predictive deep learning model.

---

2. Data Scraping

Process

- **Reddit Data**: The `praw` library is used to scrape posts from relevant subreddits like `r/stocks`, `r/investing`, and `r/wallstreetbets`. Posts are filtered based on keywords and recency.
- **Stock Data**: The `yfinance` library fetches historical stock data, including daily prices and volume.

Challenges and Resolutions

- **Reddit API Rate Limits**: Encountered rate limits while scraping a large number of posts.
    - *Solution*: Limited queries per subreddit and implemented a caching mechanism to reduce redundant requests.
- **Noise in Text Data**: Some posts contained irrelevant information or excessive formatting.
    - *Solution*: Applied preprocessing (e.g., regex cleaning) to standardize the text before sentiment analysis.

---

3. Features Extracted

Stock Features

- `Close`: The closing price of the stock.
- `Volume`: The trading volume on a given day.

Sentiment Features

- **Sentiment Score**: Average sentiment polarity of posts.
- **Post Score**: Total upvotes of relevant posts.
- **Comments Count**: Total number of comments for relevant posts.

These features were aggregated daily to align with stock price data.

---

4. Model Development

Preprocessing

- Scaled data using `MinMaxScaler` to normalize features.
- Created a sliding window of historical data (lookback = 30 days) for time-series forecasting.

Model Architecture

- **Type**: LSTM-based neural network.
- **Layers**:
  - Two LSTM layers with dropout for regularization.
  - Dense layers for final predictions.
- **Optimizer**: Adam with a learning rate of 0.001.
- **Loss Function**: Mean Squared Error (MSE).

Training

- Split the data into training (80%) and testing (20%) sets.
- Used early stopping to prevent overfitting.

---

5. Evaluation and Insights

Metrics

- **Mean Squared Error (MSE)**: Quantifies the average squared difference between predicted and actual prices.
- **Mean Absolute Error (MAE)**: Captures the average magnitude of prediction errors.

Results

- **MSE**: 35.60759
- **MAE**: 04.86289

Key Insights

- The model performs well for short-term predictions but struggles with high volatility.
- Incorporating sentiment data improved predictions compared to stock data alone.

---

6. Potential Improvements

1. **Data Integration**: Incorporate alternative sentiment sources (e.g., Twitter, news articles).
2. **Feature Engineering**: Add technical indicators like moving averages and RSI.
3. **Model Enhancement**: Explore transformer-based models for better sequential understanding.

---

7. Conclusion

This project demonstrates the feasibility of combining social media sentiment with stock data for price predictions. While the results are promising, integrating more data sources and refining the model can enhance accuracy.
