Stock Movement Prediction Using Social Media Sentiment

GithubLink : https://github.com/BusireddyHarshitha/Stock-Movement-Prediction-Using-Social-Media-Sentiment.git

Demo Video Link : https://www.youtube.com/watch?v=8zWLmZMCTaU

1. Introduction

The objective of this project was to develop a machine learning model that predicts stock movements based on sentiment analysis of discussions on social media platforms, specifically Reddit. By analyzing user-generated content from subreddits like r/stocks, the model aims to predict stock price trends using natural language processing (NLP) and machine learning techniques. In this report, I will explain the process of scraping Reddit data, the challenges encountered, the features used for stock prediction, model evaluation, and suggest potential improvements for the future.

2. Scraping Process and Challenges

2.1 Scraping Process Using PRAW

To collect relevant data, I used the Python Reddit API Wrapper (PRAW), a popular library for scraping Reddit data. I targeted subreddits such as r/stocks and r/investing to gather discussions around stock market predictions, trends, and news. PRAW allowed me to retrieve posts' titles, text, sentiment, and engagement metrics (e.g., votes, comments). The posts were scraped over a period of time, and I collected a total of 1000 recent submissions, which included both the post title and the self-text (body content).

Key components of the scraping process included:

- Initializing PRAW with API credentials: client_id, client_secret, and user_agent.
- Defining the target subreddit and setting the scraping parameters (e.g., limit of 1000 posts).
- Storing the scraped data in a structured format (pandas DataFrame) for further analysis.

2.2 Challenges and Resolutions

Several challenges were encountered during the scraping process:

1. **Handling Large Volumes of Data**: Reddit's API has rate limits to prevent overloading the server. To overcome this, I implemented a delay between requests, which allowed me to avoid hitting the API's rate limits and prevented potential bans on the API key.
2. **Missing or Incomplete Data**: Some posts were missing self-text or had incomplete titles, leading to missing values in the dataset. I addressed this by cleaning the data and filling missing values with placeholders (e.g., empty strings for missing titles or self-text).
3. **Noise in User-generated Content**: Social media data often includes irrelevant content such as spam, advertisements, and off-topic discussions. To address this, I performed text preprocessing, including removing URLs, mentions (@usernames), special characters, and stopwords.
4. **Data Structure Inconsistencies**: Reddit data often had inconsistencies in the structure of posts (e.g., some posts had long self-text, others had only titles). This was handled by focusing on the title and self-text separately for sentiment analysis, ensuring no important content was left out.

3. Features Extracted and Their Relevance to Stock Prediction

Several features were extracted from the Reddit data to assess sentiment, which could potentially influence stock price movements. These features include:

Text Features:

- **Sentiment Scores**: Using the VADER sentiment analyzer, I extracted sentiment scores for both the post titles and the body text (self-text). VADER, which is optimized for social media data, calculates a compound sentiment score ranging from -1 (negative sentiment) to +1 (positive sentiment).

- **Average Sentiment**: I combined the sentiment of both titles and self-text to calculate an average sentiment score for each post, which was used as a feature to predict stock movements.

Relevance to Stock Prediction: Sentiment scores indicate how positive or negative the community is regarding a specific stock. If a stock is discussed positively on Reddit, it might indicate a price increase, whereas negative sentiment could hint at a decline.

Engagement Features:

- **Score**: The number of upvotes or downvotes a post receives can indicate the level of interest or importance of a stock discussion.
- **Number of Comments**: Posts with high engagement (many comments) may indicate more significant discussions, potentially affecting stock movement.

Stock Data Features:

- **Historical Stock Prices**: I used yfinance to collect historical stock prices (e.g., AAPL) and merged them with the sentiment data, providing a combined dataset of stock prices and sentiments for training the prediction model.
- **Previous Sentiment (Lag Feature)**: A lag feature (previous day's sentiment) was created to account for the impact of past discussions on future stock prices.

4. Model Evaluation and Performance

4.1 Model Selection

I used various machine learning models to evaluate their performance in predicting stock movements based on the sentiment features extracted from Reddit data. The models included Random Forest, XGBoost, LightGBM, CatBoost, and more, in addition to advanced models such as LSTM (Long Short-Term Memory networks).

4.2 Model Evaluation Metrics

The performance of the models was evaluated using several metrics:

- **Accuracy**: The percentage of correct predictions.
- **Precision (0)**: Precision for predicting "no price movement" (class 0).
- **Recall (0)**: Recall for predicting "no price movement" (class 0).
- **F1-Score (0)**: F1-Score for predicting "no price movement" (class 0).
- **Precision (1)**: Precision for predicting "price movement" (class 1).
- **Recall (1)**: Recall for predicting "price movement" (class 1).
- **F1-Score (1)**: F1-Score for predicting "price movement" (class 1).

4.3 Model Performance Comparison

The performance of different models was compared using the following table:

Model	Accuracy	Precision (0)	Recall (0)	F1-Score (0)	Precision (1)	Recall (1)	F1-Score (1)
Random Forest	0.54	0.33	0.12	0.18	0.58	0.83	0.68
Random Forest with SMOTE	0.46	0.35	0.38	0.36	0.55	0.52	0.53
Modified Features (SMOTE)	0.45	0.38	0.33	0.36	0.50	0.56	0.53
LightGBM	0.42	0.36	0.33	0.34	0.47	0.50	0.49
Stacking Classifier	0.52	0.46	0.40	0.43	0.55	0.61	0.58
CatBoost	0.45	0.41	0.47	0.44	0.50	0.44	0.47
TPOT	0.55	0.50	0.40	0.44	0.57	0.67	0.62
LSTM	0.64	0.71	0.73	0.65	0.56	0.58	0.62

4.4 Insights

- **LSTM** performed the best in terms of accuracy and F1-score for predicting "price movement" (class 1), with an accuracy of 64%. It also showed a strong performance in recall (73%) and F1-score (65%) for positive price movements.
- **XGBoost** also performed well, achieving 58% accuracy, with good precision (55%) and recall (72%) for predicting price increases.
- **Random Forest** showed moderate performance with an accuracy of 54% but had a lower F1-score for "no price movement" (class 0).

5. Suggestions for Future Expansions

Incorporating Multiple Data Sources:

- **Twitter**: Twitter provides real-time stock-related tweets from a large user base, which could offer more immediate insights into stock movements.
- **Telegram Channels**: Telegram has private groups and channels with more focused stock discussions, which could help refine predictions.
- **News Articles**: News articles about stocks could be another valuable data source, providing a broader context for discussions.

Advanced Natural Language Processing:

- **Deep Learning Models**: Using advanced NLP models like BERT or GPT-3 could help capture complex relationships between stock discussions and movements.
- **Topic Modeling**: Applying topic modeling (e.g., LDA) can help identify specific topics being discussed, allowing the model to focus on stock-specific news or trends.

Feature Engineering:

- **Incorporating Financial Indicators**: Additional stock-specific features like trading volume, price volatility, and technical indicators (e.g., RSI, MACD) could improve the model's predictive power.

- **Time-Series Analysis**: Time-series models (e.g., ARIMA, LSTM) could be integrated with the sentiment data to forecast stock price movements more accurately.

Model Tuning and Optimization:

- Hyperparameter tuning for the Random Forest model or trying other classifiers (e.g., XGBoost, SVM) could improve the model's accuracy.
- Fine-tuning the sentiment analysis process (e.g., adjusting the sentiment thresholds or using domain-specific lexicons) could lead to better feature extraction and improved predictions.

6. Conclusion

This project successfully highlighted the potential of using social media sentiment, specifically from Reddit discussions, to predict stock movements. By analyzing sentiment from subreddits like r/stocks and r/investing, we observed that the collective sentiment of users could offer valuable insights into stock price trends. Positive sentiment often aligned with potential price increases, while negative sentiment could indicate declines, making sentiment analysis a promising approach for stock prediction. The model, especially using LSTM, achieved an encouraging accuracy of 64%, demonstrating its effectiveness in leveraging social media data for financial forecasting.

The performance of the model can be further enhanced by integrating additional data sources, such as Twitter or financial news, and employing advanced NLP techniques like BERT to better capture the complexities of social media language. Additionally, incorporating stock-specific features like technical indicators and trading volume can refine the model's predictions and improve its overall accuracy.

Looking ahead, expanding the model to include diverse data sources and further optimizing it holds great potential for creating a powerful tool for investors. By continuously analyzing real-time market sentiment alongside traditional financial data, this system can provide valuable insights, helping investors make more informed and timely decisions.