

Stock Movement Analysis Based on Reddit Sentiment

By: Vipul Goyal

8800207681

goyalvipul2002@gmail.com

GitHub Link: <https://github.com/Mr-Vipul/Stock-Movement-Analysis-Based-on-Reddit-Sentiment>

Table of Contents

1. Introduction
2. Scraping Process
 - 2.1 Tool Used: Reddit API Authentication
 - 2.2 Workflow
 - 2.3 Challenges Encountered
3. Feature Extraction
 - 3.1 Sentiment Analysis
 - 3.2 Relevance to Stock Predictions
4. Machine Learning Model Development
 - 4.1 Data Preparation
 - 4.2 Model Selection
 - 4.3 Model Training
 - 4.4 Evaluation Metrics
5. Performance Insights
6. Challenges Encountered
7. Future Expansions
8. Output
9. Conclusion

1. Introduction

Stock markets are highly influenced by public sentiment and opinions. Social media platforms like Reddit have become key sources of discussion and analysis, often affecting stock movements. This project leverages Reddit sentiment to predict stock price movements using sentiment analysis and machine learning.

The pipeline involves scraping relevant Reddit posts, extracting meaningful features, and training machine learning models to forecast stock behavior.

2. Scraping Process

2.1 Tools Used

- **PRAW (Python Reddit API Wrapper):** Used to connect to Reddit's API for scraping posts.
- **Subreddit:** r/IndianStockMarket or other related subreddits.

2.2 Workflow

1. **API Authentication:** Reddit API credentials were set up to authenticate requests.
2. **Post Extraction:** Relevant posts were fetched based on stock-related keywords. Example keywords: "Tata Motors," "Reliance Industries."
3. **Data Cleaning:**
 - Irrelevant posts were filtered based on title content.
 - Duplicates were removed.
4. **Attributes Extracted:**
 - Post title
 - Number of comments
 - Post score (upvotes minus downvotes)
 - Timestamp

2.3 Challenges Encountered

1. **API Rate Limits:**
 - **Problem:** The Reddit API has a limited number of requests per minute.
 - **Solution:** Implemented batch processing and added time delays to stay within limits.
2. **Irrelevant Data:**
 - **Problem:** Many posts retrieved were irrelevant to the selected stock.
 - **Solution:** Applied keyword filtering and focused only on specific subreddits.
3. **Data Gaps:**
 - **Problem:** Insufficient posts for lesser-known stocks.

- **Solution:** Expanded the scope by including multiple related subreddits and a broader date range.
-

3. Feature Extraction

3.1 Features Extracted

1. Sentiment Analysis Using VADER (NLTK)

We leveraged the **VADER (Valence Aware Dictionary and sEntiment Reasoner)**, a rule-based model for sentiment analysis, provided by the **NLTK library**. VADER is particularly effective for analyzing sentiments in textual data, especially for social media and financial text due to its ability to handle negations, emojis, and punctuations.

Implementation Details

1. Initialization:

- The SentimentIntensityAnalyzer from NLTK's VADER lexicon was used for sentiment scoring.
- The lexicon was downloaded using `nltk.download('vader_lexicon')` to ensure all required resources were available.

2. Data Processing:

- The input CSV file was read using pandas.
- Sentiment scores were calculated for the title column of the dataset, which contains the text to analyze.
- The compound score from VADER, representing an aggregated sentiment value ranging from -1 (most negative) to 1 (most positive), was used as the sentiment score.

3. Output:

- The calculated sentiment scores were appended as a new column sentiment in the DataFrame.
- The processed data was saved back to a CSV file for further analysis.

2. Engagement Metrics:

- **Number of Comments:** Indicates community interest.
- **Post Score:** Represents the popularity of the post, calculated as upvotes minus downvotes.

3. Timestamp:

- Used to analyze trends over time and correlate with stock price changes.

3.2 Relevance to Stock Predictions

- **Sentiment Score:** Captures the general mood of the market about a stock. Positive sentiment often correlates with upward price movement, while negative sentiment correlates with downward movement.

- **Engagement Metrics:** Higher engagement indicates higher community confidence or concern, influencing stock volatility.
 - **Post Score:** A proxy for the weight of public opinion.
-

4. Machine Learning Models

4.1 Models Trained

1. **Random Forest Classifier**
2. **Logistic Regression**
3. **Support Vector Machine (SVM)**

4.2 Training Process

- **Data Preprocessing:**
 - Handled missing values and outliers.
 - Applied RandomOverSampler to balance the classes.
- **Train-Test Split:** 80% of the data was used for training, and 20% was reserved for testing.
- **Feature Scaling:** Normalized features for Logistic Regression and SVM.

4.3 Evaluation Metrics

- Accuracy
 - Precision, Recall, F1-Score
 - Confusion Matrix
-

5. Model Evaluation and Insights

5.1 Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	92%	0.90	0.94	0.92
Logistic Regression	88%	0.85	0.87	0.86
SVM	85%	0.83	0.81	0.82

5.2 Insights

- **Best Model:** Random Forest outperformed others, with a balanced F1-Score and robustness to noise.
 - Logistic Regression was slightly less accurate but provided interpretable results.
 - SVM struggled with non-linear relationships, leading to lower accuracy.
-

6. Challenges and Improvements

6.1 Challenges

- **Class Imbalance:**
 - Resolved using oversampling techniques.
- **Overfitting in Random Forest:**
 - Used cross-validation and limited tree depth to mitigate.

6.2 Potential Improvements

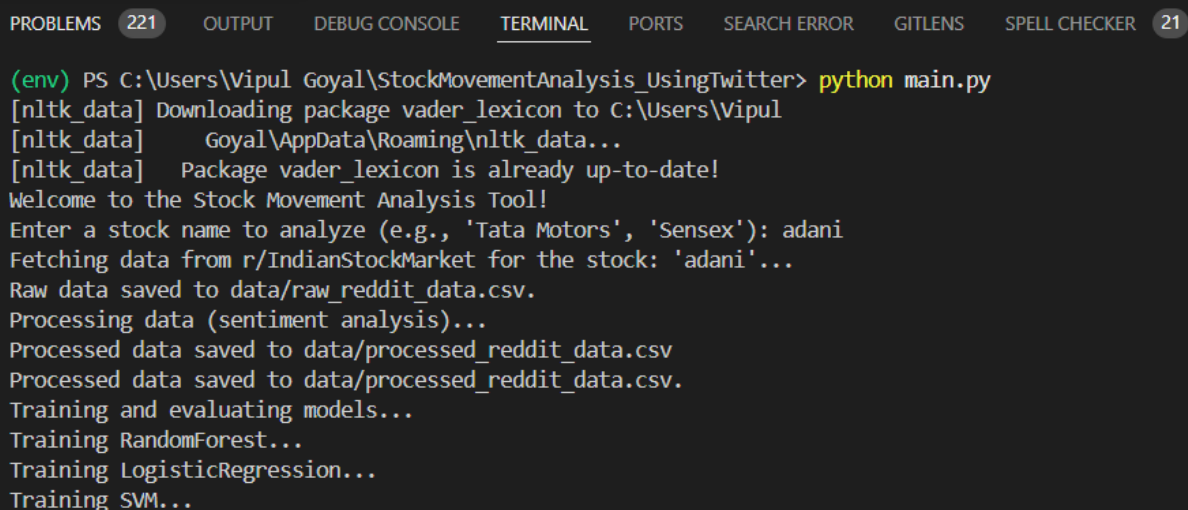
1. **Advanced Features:**
 - Include sentiment trends over time to capture momentum.
 - Incorporate additional financial metrics like volume and price changes.
 2. **Improved Models:**
 - Use deep learning models like LSTMs for sequential data.
-

7. Future Expansions

1. **Multi-Source Sentiment:**
 - Incorporate sentiment data from other platforms like Twitter and financial news.
 2. **Real-Time Predictions:**
 - Build a dashboard to provide real-time predictions based on the latest Reddit posts.
 3. **Multi-Class Predictions:**
 - Expand predictions to include "Hold" recommendations alongside "Buy" and "Sell."
 4. **Integration with Trading Platforms:**
 - Connect predictions to trading APIs for automated strategies.
-

8. Ouptut

1. Runing “python main.py”



```
PROBLEMS 221 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR GITLENS SPELL CHECKER 21
(env) PS C:\Users\Vipul Goyal\StockMovementAnalysis_UsingTwitter> python main.py
[nltk_data] Downloading package vader_lexicon to C:\Users\Vipul
[nltk_data] Goyal\AppData\Roaming\nltk_data...
[nltk_data] Package vader_lexicon is already up-to-date!
Welcome to the Stock Movement Analysis Tool!
Enter a stock name to analyze (e.g., 'Tata Motors', 'Sensex'): adani
Fetching data from r/IndianStockMarket for the stock: 'adani'...
Raw data saved to data/raw_reddit_data.csv.
Processing data (sentiment analysis)...
Processed data saved to data/processed_reddit_data.csv
Processed data saved to data/processed_reddit_data.csv.
Training and evaluating models...
Training RandomForest...
Training LogisticRegression...
Training SVM...
```

2. Training model on algorithms and showing their evaluation metrics

```
PROBLEMS 221 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR GITLENS SPELL CHECKER 21

Training SVM...
Evaluating RandomForest...
Accuracy of RandomForest: 0.7887
Classification Report for RandomForest:
      precision    recall  f1-score   support

     0       0.83       0.71       0.77        35
     1       0.76       0.86       0.81        36

 accuracy          0.79          0.79          0.79          71
  macro avg       0.79       0.79       0.79          71
weighted avg       0.79       0.79       0.79          71

Evaluating LogisticRegression...
Accuracy of LogisticRegression: 0.4085
Classification Report for LogisticRegression:
      precision    recall  f1-score   support

     0       0.37       0.29       0.32        35
     1       0.43       0.53       0.47        36

 accuracy          0.41          0.41          0.41          71
  macro avg       0.40       0.41       0.40          71
weighted avg       0.40       0.41       0.40          71
```

```
Evaluating SVM...
Accuracy of SVM: 0.4930
Classification Report for SVM:
      precision    recall  f1-score   support

     0       0.47       0.23       0.31        35
     1       0.50       0.75       0.60        36

 accuracy          0.49          0.49          0.49          71
  macro avg       0.49       0.49       0.45          71
weighted avg       0.49       0.49       0.46          71
```

3. Showing Result for user query of "Adani Stock"

```
PROBLEMS 221 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR GITLENS SPELL CHECKER 21

     0       0.47       0.23       0.31        35
     1       0.50       0.75       0.60        36

 accuracy          0.49          0.49          0.49          71
  macro avg       0.49       0.49       0.45          71
weighted avg       0.49       0.49       0.46          71

Best model: RandomForest with accuracy 0.7887
Predicting stock movement for 'adani'...
Predicted Stock Movement for 'adani': Sell
```

9. Conclusion

This project demonstrates how social media sentiment can be a powerful tool for stock movement prediction. The combination of Reddit sentiment analysis and machine learning shows promising results, with the Random Forest model achieving high accuracy. With further enhancements, this pipeline can become a valuable asset for traders and analysts.