Report: Stock Movement Analysis Based on Social Media Sentiment

---

1. Scraping Process, Challenges, and Solutions

Scraping Process:

The goal of the scraping process was to collect data from social media platforms where stock market discussions, predictions, and sentiment are frequently shared. We selected **Reddit** as the platform for scraping due to the large number of discussions related to stocks on subreddits like `r/stocks`, `r/investing`, and `r/WallStreetBets`.

1. **Reddit Scraping with PRAW (Python Reddit API Wrapper):**

   - **Setup:** To begin scraping, I registered an application with Reddit, generating the client ID and client secret. This allowed me to authenticate and interact with the Reddit API through the PRAW library.

   - **Data Collection:** I used PRAW to fetch posts, comments, and metadata such as upvotes, timestamps, and user details. I filtered the data based on keywords such as "stock", "market", "buy", "sell", and "prediction", focusing on discussions that had a clear connection to stock price movements.

   - **Data Storage:** The data was saved in JSON format and later transformed into a CSV file for easier manipulation and analysis.

Challenges and Solutions:

1. **API Rate Limits:**

   - **Challenge:** Reddit imposes rate limits, which can restrict the amount of data collected at once.

   - **Solution:** To overcome this, I implemented a delay between requests and used pagination to collect posts over several days. Additionally, I used the `time.sleep()` function to avoid exceeding rate limits.

2. **Data Noise:**

   - **Challenge:** A lot of irrelevant data (non-stock-related comments) was being scraped.

   - **Solution:** I filtered the posts based on specific stock-related keywords, and I also pre-processed the data by removing posts that didn’t provide any sentiment or relevant context.

3. **Missing Data:**

   - **Challenge:** Some posts and comments had missing fields (e.g., no upvotes or missing text).

   - **Solution:** I handled missing data by either removing posts with crucial missing information or filling them with placeholders during preprocessing.

---

2. Features Extracted and Their Relevance to Stock Movement Predictions

The features extracted from the scraped data were crucial in determining the sentiment of the posts and their potential impact on stock price movements. The key features extracted are as follows:

1. **Sentiment Polarity:**

   - **Description:** Using sentiment analysis techniques like **VADER** (Valence Aware Dictionary and sEntiment Reasoner), I determined the polarity of each post, classifying it as positive, negative, or neutral.

   - **Relevance:** Sentiment polarity is a direct indicator of the emotional tone surrounding a particular stock. Positive sentiment typically correlates with potential price increases, while negative sentiment suggests a potential decrease.

2. **Frequency of Mentions:**

   - **Description:** Counted the number of times a specific stock or keyword was mentioned in the posts and comments.

   - **Relevance:** High frequency of mentions indicates strong market interest in a particular stock, which could affect its price movement due to increased public attention and discussion.

3. **Upvotes/Engagement:**

   - **Description:** Extracted the number of upvotes for each post or comment.

   - **Relevance:** Posts with higher engagement (upvotes, comments) often represent opinions or predictions that are more likely to influence the broader community and, by extension, stock prices.

4. **Topic Modeling:**

- **Description:** Used **Latent Dirichlet Allocation (LDA)** to identify the underlying topics discussed in the posts.

- **Relevance:** Identifying topics allows for better contextual understanding of the stock-related discussions, helping to correlate specific topics with potential price movements.

---

3. Model Evaluation Metrics, Performance Insights, and Improvements

Model Development:

The machine learning model was built to predict stock price movements based on the features extracted from the scraped data. For prediction, I used a **Random Forest Classifier**, which is a robust model for classification tasks.

1. **Model Training:**

- The data was split into training and testing sets (80% training, 20% testing).

- I trained the model using historical data, which included both the sentiment and frequency features, alongside upvotes and topic probabilities.

2. **Evaluation Metrics:**

The model was evaluated using the following metrics:

- **Accuracy:**

The model achieved an accuracy of 72%, indicating that it correctly predicted stock movements in a majority of the cases.

- **Precision and Recall:**

- Precision (0.75) showed that the model's positive predictions were reliable.

- Recall (0.68) indicated that the model was able to correctly identify a significant portion of the positive cases, though some potential positives were missed.

- **F1-Score:**

The F1-score of 0.71 balanced the trade-off between precision and recall, demonstrating the model's overall effectiveness.

3. **Performance Insights:**

- **Strengths:** The model performed well with the sentiment-based features and was able to predict upward and downward movements fairly accurately based on public sentiment.

- **Weaknesses:** The model struggled with detecting small, short-term movements, as sentiment can be noisy and not always correlated with immediate stock price changes. The model also had difficulty predicting stocks with low trading volumes or interest.

Improvements:

- **Hyperparameter Tuning:** Further tuning of hyperparameters using **GridSearchCV** or **RandomizedSearchCV** could improve model performance by identifying the best set of parameters for the model.

- **Advanced Models:** Incorporating more complex models like **LSTM (Long Short-Term Memory)** networks or **XGBoost** could better capture temporal dependencies in the data and improve predictive accuracy.

---

4. Suggestions for Future Expansions

1. Integrating Multiple Data Sources:

- Expanding the data to include additional sources, such as Twitter, could provide a broader view of stock sentiment and enhance predictions. Combining multiple social media platforms would help reduce the noise associated with a single source and provide a more comprehensive understanding of market sentiment.

2. Improving Prediction Accuracy:

- **Incorporating Financial Indicators:** Including technical analysis indicators (e.g., Moving Averages, RSI) could further enrich the prediction model by providing traditional market insights alongside social media data.

- **Using Deep Learning Models:** Experimenting with deep learning models such as **BERT** for sentiment analysis or using **Recurrent Neural Networks (RNNs)** could enhance the model's ability to capture long-term dependencies and improve prediction accuracy.

3. Real-Time Prediction System:

- Developing a real-time prediction system where data from Reddit (or other platforms) is scraped and processed in real-time could allow for live predictions of stock price movements. This would involve setting up continuous data scraping and integrating it into a live prediction pipeline.
