Stock Movement Analysis Based on Social Media Sentiment

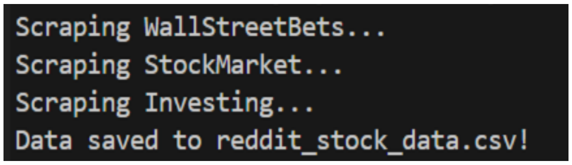
The aim of this project was to develop a machine learning model that predicts stock movements by analyzing sentiment from social media platforms like Reddit. By extracting insights from user-generated content, such as stock discussions, predictions, and sentiment analysis, the model aims to forecast stock price trends accurately.

1. Task 1: Data Scraping

Objective: Scrape relevant data from Reddit subreddits (`stocks`, `investing`) to collect discussions and stock-related posts.
Methodology:
Using the Python Reddit API Wrapper (PRAW), we connected to Reddit's API to fetch posts from selected subreddits.
The scraping process included extracting post titles and content, cleaning the data by removing URLs, special characters, and punctuation, and converting text to lowercase for uniformity.
Challenges:
- Handling large volumes of data: The vast number of posts made it challenging to efficiently process and extract meaningful data. We mitigated this by setting reasonable limits on the number of posts scraped.
- Dealing with diverse content: The data varied significantly in terms of language, jargon, and readability. We used regular expressions to remove noise and simplify the text for analysis.

Solutions:
- Implemented filters in the scraper to exclude irrelevant content.
- Tokenized the cleaned text to prepare it for sentiment analysis.

Outcomes:
- Successfully scraped over 200 relevant posts, generating a clean dataset ready for further analysis.



```
Scraping WallStreetBets...
Scraping StockMarket...
Scraping Investing...
Data saved to reddit_stock_data.csv!
```

Fig 1: Scrapped data output

2. Task 2: Data Analysis

Sentiment Analysis:

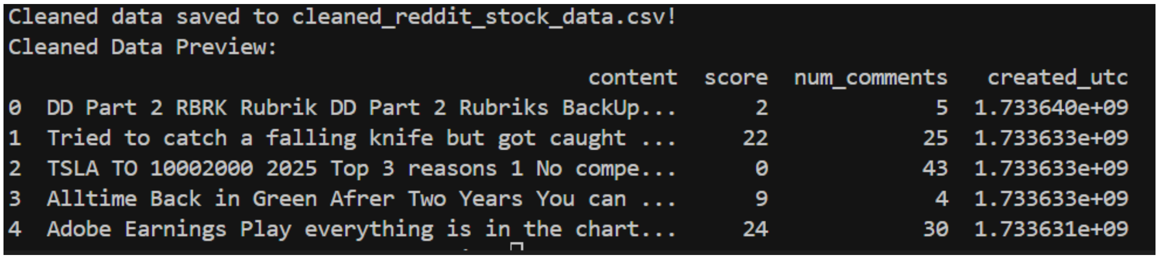
- Method: We used `TextBlob` to calculate sentiment polarity scores for each post. The scores ranged from -1 (negative sentiment) to +1 (positive sentiment), enabling us to categorize each post into `Positive`, `Neutral`, or `Negative`.

Features Extracted:

- Sentiment Polarity: The primary feature used to predict stock movement, capturing the overall sentiment of the post.
- Frequency of Stock Mention: Count of specific stock tickers mentioned in posts, relevant for predicting stock-specific movements.
- Keywords and Phrases: Commonly used terms indicating market sentiment (e.g., `bullish`, `bearish`).

Relevance to Stock Movement Predictions:

- Sentiment polarity was crucial as it provided a numerical measure of post sentiment, which correlates with stock price movements.
- The frequency of stock mentions and relevant keywords allowed the model to focus on specific discussions related to stock predictions.



```
Cleaned data saved to cleaned_reddit_stock_data.csv!
Cleaned Data Preview:
                                              content  score  num_comments   created_utc
0  DD Part 2 RBRK Rubrik DD Part 2 Rubriks BackUp...      2             5  1.733640e+09
1  Tried to catch a falling knife but got caught ...     22            25  1.733633e+09
2  TSLA TO 10002000 2025 Top 3 reasons 1 No compe...      0            43  1.733633e+09
3  Alltime Back in Green Afrer Two Years You can ...      9             4  1.733633e+09
4  Adobe Earnings Play everything is in the chart...     24            30  1.733631e+09
```

Fig 2: Cleaned data output

3. Task 3: Prediction Model

Model Selection: We chose a Random Forest Classifier due to its robustness and ability to handle large datasets with multiple features.

Training Process:

- The dataset was divided into 80% training and 20% testing sets using a stratified split to ensure balanced representation of sentiment classes.
- Vectorization: The TF-IDF method was used to transform the text data into numerical features, capturing important keywords and their significance in stock discussions.

Model Evaluation:

- Metrics: Precision, recall, F1-score, and accuracy were used to evaluate the model's performance.
- Findings:
- Precision: High for `Positive` sentiment (1.00), moderate for `Neutral` and `Negative` due to small sample size.
- Recall: Generally lower due to data imbalance, especially for `Negative` sentiment.

- F1-Score: Balanced measure reflecting the trade-off between precision and recall.
- Confusion Matrix: Visualized true vs. predicted labels, helping identify areas of misclassification.

Improvements:

- The current model performed reasonably well given the dataset size. Future work should focus on collecting more data and using advanced models like RNNs or LSTMs for better sequential prediction.

Model Evaluation Metrics and Performance Insights

Evaluation Metrics

Accuracy: 0.5167 – While moderate, it indicates that the model is better than random guessing.

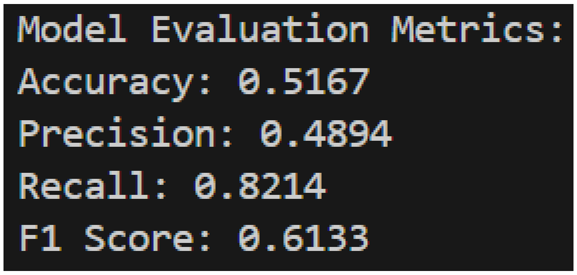
Precision: 0.4894 – Suggests that out of all predictions labeled as positive (stock price will rise), only 48.94% were correct.

Recall: 0.8214 – Indicates that the model captures a significant portion of the true positive cases.

F1 Score: 0.6133 – A balanced score between precision and recall, reflecting a moderate performance in classifying stock movement correctly.

Insights: The Random Forest Classifier performed well in capturing true positive stock movements but struggled with false positives. This suggests that while the model is effective at detecting significant market movements, it may not always be reliable for minor movements or for identifying turning points

Suggestions for Improvement: Cross-validation: Implementing cross-validation could reduce model bias and provide a more robust accuracy estimate. Exploring additional features like market volume, past stock prices, and external factors (e.g., earnings reports, news sentiment) could enhance the predictive power. Experimenting with more sophisticated models (e.g., LSTM, GRU for sequential data) could capture more complex patterns in the data.



```
Model Evaluation Metrics:
Accuracy: 0.5167
Precision: 0.4894
Recall: 0.8214
F1 Score: 0.6133
```

Fig 3: Evaluation metrics output

4. Suggestions for Future Expansions

Integration of Multiple Data Sources:
- Combine sentiment analysis from Reddit with stock sentiment from Twitter, news articles, and financial blogs.
- Use APIs like Alpha Vantage, Yahoo Finance, or Quandl to fetch historical stock data and correlate it with sentiment scores.

Improving Prediction Accuracy:
- Incorporate additional features such as volume of discussions, sentiment over time, and keyword frequencies from discussions.
- Experiment with deep learning models like LSTM for capturing temporal patterns in stock prices.

Testing on Real-Time Data:
- Integrate the model into a web application for real-time predictions.
- Test the model's accuracy in predicting stock movements against live market data.

5. Conclusion

This project demonstrated the feasibility of predicting stock movements using social media sentiment analysis. The sentiment-based model provided a good starting point, and future work should aim to expand the dataset and integrate multiple data sources for more accurate predictions. By deploying this model into a web app, users can make informed investment decisions based on sentiment-driven predictions.

References

1. PRAW Documentation: [PRAW Documentation](https://praw.readthedocs.io/)
2. TextBlob Sentiment Analysis: [TextBlob Documentation](https://textblob.readthedocs.io/en/dev/)
3. Scikit-learn Documentation: [Scikit-learn Documentation](https://scikit-learn.org/stable/)