Title: Project Type 1 (Social Media Analysis of Stock Market) Reddit

Authors

Geoffrey Grossthal

Gage Smith

Executive Summary

1. Abstract
2. Introduction
3. Related Works
    a. A review of previous research or studies related to the topic.
    b. Discussion of key findings from other sources that inform the current study.
4. Model Description
    a. A detailed explanation of the model or methodology used in the study or analysis.
    b. Describes the techniques, algorithms, or approaches employed to address the research question.
5. Experiment
    a. A description of the experimental setup, including data collection, tools used, and procedures.
    b. Results of the experiments or analysis, often presented with charts, tables, and statistical data.
6. Future Works
    a. Suggestions for future research or improvements.
    b. Discussion of limitations of the current study and areas that could be expanded or explored further.
7. References
    a. A list of all the sources, studies, articles, and resources cited throughout the report.

Abstract

A stock trading strategy can be developed by combining sentiment analysis from Reddit using PRAW (Python Reddit API Wrapper) and BERT (Bidirectional Encoder Representations from Transformers) with historical stock data from Yahoo Finance (YFinance). Reddit discussions, particularly from subreddits like r/wallstreetbets, provide real-time market sentiment, which can be analyzed with BERT to classify posts and comments as bullish or bearish. These sentiment signals can then be used to inform trading decisions, such as buying or selling a stock. At the same time, YFinance offers crucial financial data, such as stock prices and technical indicators, to complement sentiment-driven signals. By integrating both sentiment analysis and technical analysis, a more holistic trading algorithm can be created, allowing for

better-informed buy/sell decisions. This strategy can be backtested on historical data to optimize performance, making it a promising approach to enhance stock trading by incorporating social media sentiment into traditional financial analysis.

Introduction

In this project, we aim to develop a trading strategy by leveraging both Reddit sentiment analysis and stock market data. The core of the strategy is built around analyzing user-generated content from Reddit to gauge the collective sentiment towards specific stocks. Using the VADER SentimentIntensityAnalyzer, which is a lexicon-based tool designed for analyzing the sentiment of short, informal text (like Reddit posts), we classify posts as positive, negative, or neutral. This sentiment score is then used to inform trading decisions. Additionally, for more sophisticated sentiment analysis in the financial domain, we incorporate FinBERT, a model specifically trained to understand financial language. When the model type is set to finbert, we use the following code to initialize the model: ProsusAI/finbert, which accesses the FinBERT model from Hugging Face. By combining insights from both VADER and FinBERT, we enhance our analysis and develop more robust strategies for predicting stock movements based on market sentiment and trends. Ultimately, this framework allows us to backtest and implement data-driven trading strategies using sentiment data from Reddit alongside historical stock prices.

Related Works

The study "Predicting $GME Stock Price Movement Using Sentiment from Reddit" by Wang and Luo (2021) closely aligns with the current project, as both aim to use sentiment analysis from Reddit discussions to predict stock price movements. Specifically, the authors focused on r/wallstreetbets, the very community that heavily influenced the dramatic rise in GameStop's (GME) stock price in early 2021. Like this project, the authors used VADER for sentiment analysis and explored semantic models like word2vec and BERT to capture the community’s sentiment, which they hypothesized would correlate with GME’s price movements. Their work reinforces the idea that social media platforms, particularly Reddit, can provide valuable insights into market sentiment and can be used as a predictive tool for stock price fluctuations, a concept central to the current project’s approach to integrating sentiment data with stock trading strategies. The unique linguistic features and irony within r/wallstreetbets are similarly considered in this project, enhancing the relevance of their findings to the current research.

Model Description

PrawProxy.py

This script leverages the Reddit API Wrapper (praw) to collect, filter, and organize Reddit posts related to the Apple Inc. stock (AAPL) based on predefined keywords. It authenticates using credentials from a configuration file (PrawConfig.json) and retrieves posts from specified subreddits, applying filters for time intervals and relevance. Using regex-based keyword matching, the script identifies posts discussing AAPL and serializes them into a custom data model (RedditPost). Relevant posts are then stored as JSON files in a structured directory organized by stock ticker, year, month, and day. Duplication checks ensure that posts are not saved redundantly, and the run_data_collection function automates the process across multiple subreddits and time filters. This tool is designed for monitoring financial discussions and preparing data for further analysis.

RedditPost.py

This file defines the RedditPost class, a data model used to structure and standardize Reddit post information for further processing. Each RedditPost instance encapsulates attributes such as the post's title, score, URL, content, creation timestamp, subreddit name, and the number of comments. The class provides utility methods like to_dict for converting the object into a dictionary format, suitable for JSON serialization, and to_string for generating a human-readable representation. This model serves as a foundational component for organizing and handling Reddit data in a consistent and extensible manner, facilitating integration with other parts of the data collection and storage pipeline.

analyze_live_data.py

This script conducts live sentiment analysis on Reddit discussions about stocks by processing data stored in organized folders. It uses the StockSentimentAnalyzer class with the VADER model to evaluate daily sentiment scores for each stock ticker. For each ticker, the script computes metrics such as average sentiment, total posts analyzed, and identifies the most positive and negative days. The results are saved as CSV files and visualized with line plots, providing a clear picture of sentiment trends over time. Additionally, an overall summary is generated, consolidating key insights across all analyzed tickers. This tool facilitates real-time tracking of public sentiment, offering valuable insights for financial analysis.

GatherSentiment.py

This script integrates live and historical sentiment analysis of Reddit posts to generate actionable insights for a specified stock ticker, focusing on AAPL by default. It facilitates fetching recent and historical Reddit data, analyzing the sentiment of posts using the VADER sentiment analyzer, and calculating a sentiment score based on the ratio of positive and negative posts. The script also supports updating a comprehensive historical dataset of sentiment scores, enabling backtesting and trend analysis. Results are stored in structured JSON files, with daily sentiment scores organized chronologically. Additionally, users can choose to update live data or work with existing records, allowing for flexibility in real-time and retrospective sentiment analysis.

StockSentimentAnalyzer.py

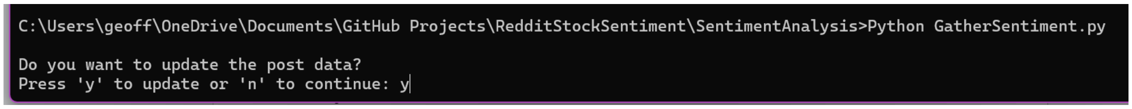
This script provides a framework for analyzing the sentiment of Reddit posts associated with stock tickers. It supports two sentiment analysis models: VADER (rule-based) and FinBERT (fine-tuned on financial data). Posts are loaded from JSON files within a data directory, combined into a DataFrame, and analyzed for sentiment. The script calculates a weighted daily sentiment score for each stock ticker, factoring in the score and comment count of posts to prioritize highly engaged content. Results include metrics like average compound sentiment, positive and negative post counts, and a weighted sentiment score, enabling deeper insights into market sentiment trends.

TradingStrategy.py

Script provides a comprehensive framework for backtesting sentiment-based stock trading strategies. It uses sentiment scores, which can be derived from sources like Reddit posts or financial models, to generate buy, sell, or hold signals. The strategy allows for customizable thresholds, enabling the user to control the level of sentiment required to trigger trading actions. The script downloads historical stock data using Yahoo Finance and merges it with sentiment data to simulate trades over a specified date range. It tracks portfolio performance, calculating key metrics such as return on investment (ROI), Sharpe ratio, and maximum drawdown. The results are visualized in a portfolio value graph, showing buy and sell points, and the performance of different stocks can be compared. The script can also analyze various threshold combinations to optimize trading decisions. Finally, it generates a summary report and optional correlation analysis for multiple stocks, making it a powerful tool for testing and refining sentiment-based trading strategies.

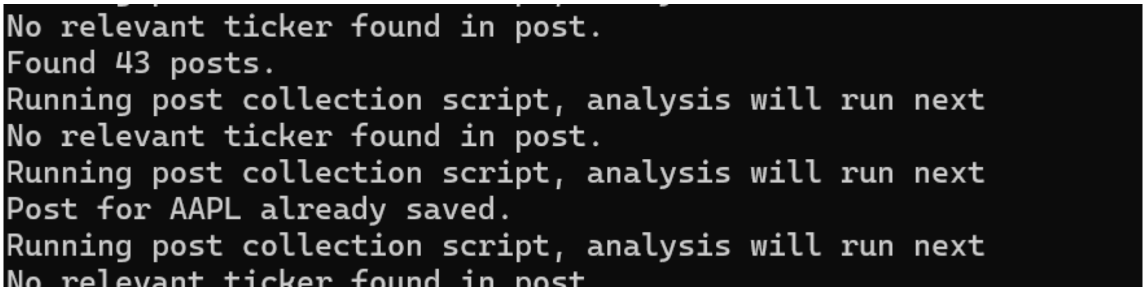
Experiment

1. GatherSentiment.py
    a. Updated the post data section with recent posts



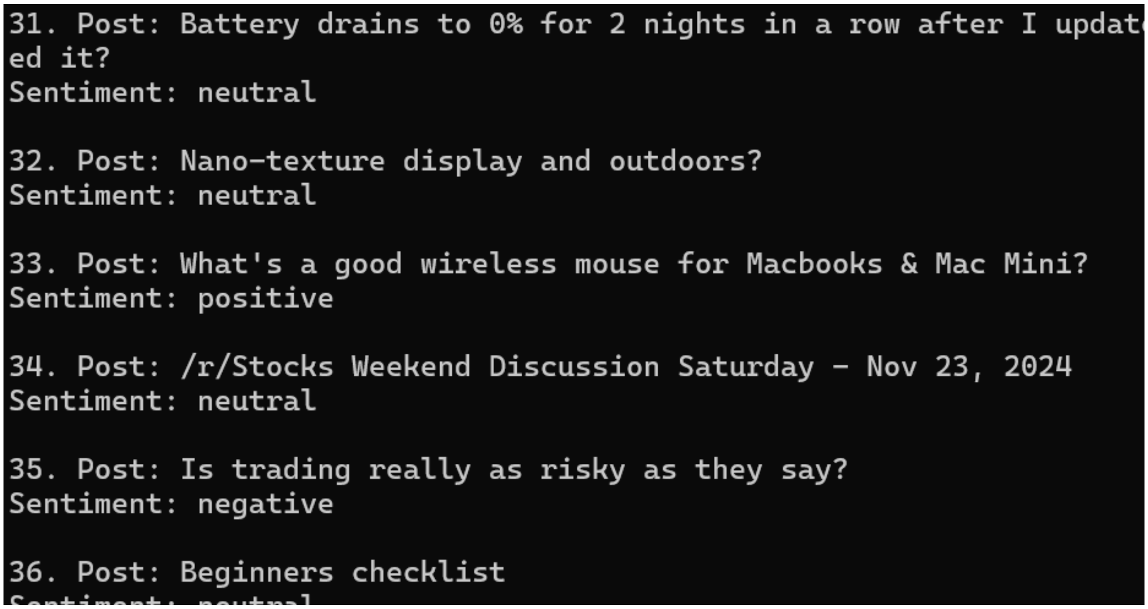
```
C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis>Python GatherSentiment.py

Do you want to update the post data?
Press 'y' to update or 'n' to continue: y
```

Data will be collected and appended to appropriate folders



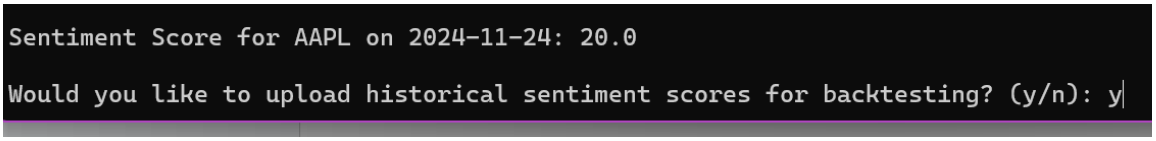
```
No relevant ticker found in post.
Found 43 posts.
Running post collection script, analysis will run next
No relevant ticker found in post.
Running post collection script, analysis will run next
Post for AAPL already saved.
Running post collection script, analysis will run next
No relevant ticker found in post
```

Sentiment scores will be generated for individual post



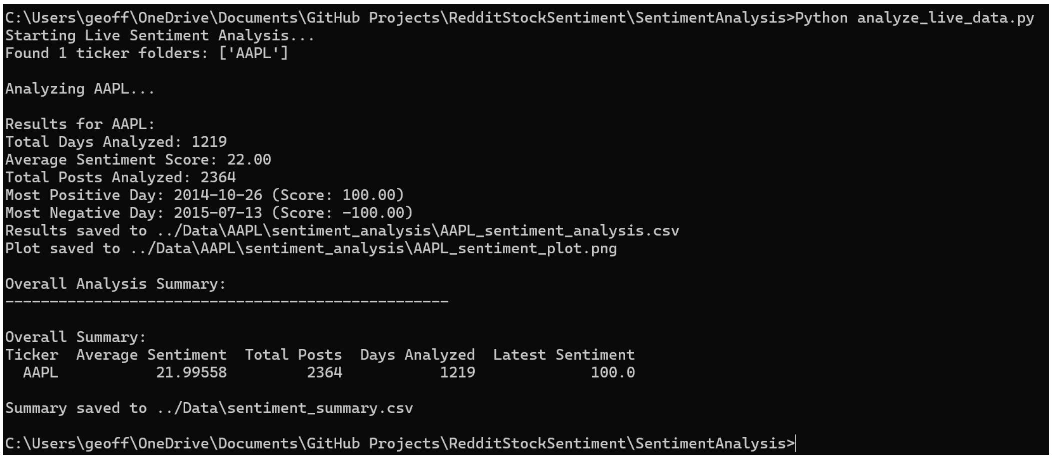
```
31. Post: Battery drains to 0% for 2 nights in a row after I update
ed it?
Sentiment: neutral

32. Post: Nano-texture display and outdoors?
Sentiment: neutral

33. Post: What's a good wireless mouse for Macbooks & Mac Mini?
Sentiment: positive

34. Post: /r/Stocks Weekend Discussion Saturday - Nov 23, 2024
Sentiment: neutral

35. Post: Is trading really as risky as they say?
Sentiment: negative

36. Post: Beginners checklist
Sentiment: neutral
```

Sentiment score will be given for the current day by the given formula with option for it to be uploaded for back testing



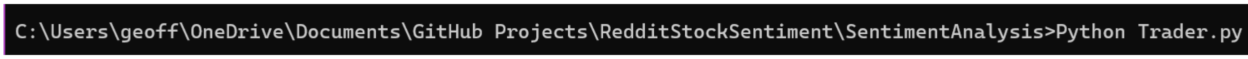
```
Sentiment Score for AAPL on 2024-11-24: 20.0

Would you like to upload historical sentiment scores for backtesting? (y/n): y
```

Gather live data

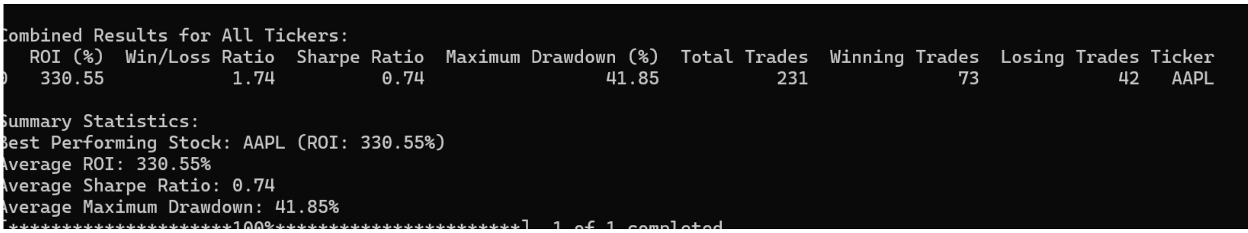


```
C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis>Python analyze_live_data.py
Starting Live Sentiment Analysis...
Found 1 ticker folders: ['AAPL']

Analyzing AAPL...

Results for AAPL:
Total Days Analyzed: 1219
Average Sentiment Score: 22.00
Total Posts Analyzed: 2364
Most Positive Day: 2014-10-26 (Score: 100.00)
Most Negative Day: 2015-07-13 (Score: -100.00)
Results saved to ../Data\AAPL\sentiment_analysis\AAPL_sentiment_analysis.csv
Plot saved to ../Data\AAPL\sentiment_analysis\AAPL_sentiment_plot.png

Overall Analysis Summary:
--------------------------------------------------

Overall Summary:
Ticker  Average Sentiment  Total Posts  Days Analyzed  Latest Sentiment
  AAPL           21.99558         2364           1219             100.0

Summary saved to ../Data\sentiment_summary.csv

C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis>
```

Gather trading metrics and statistics

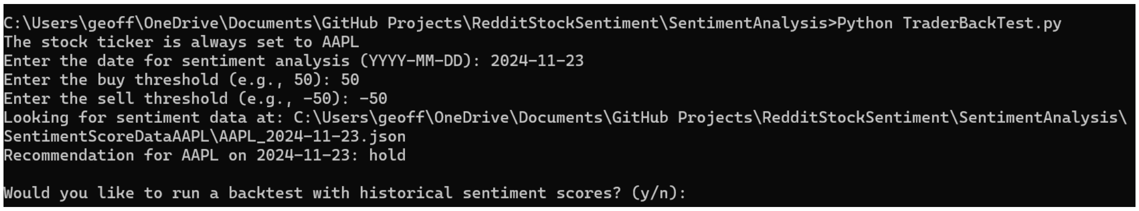


```
C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis>Python Trader.py
```



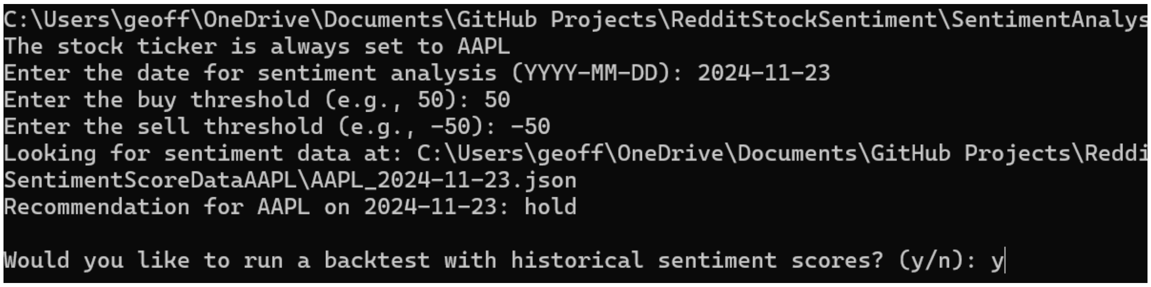
```
Combined Results for All Tickers:
   ROI (%)  Win/Loss Ratio  Sharpe Ratio  Maximum Drawdown (%)  Total Trades  Winning Trades  Losing Trades Ticker
0   330.55            1.74          0.74                 41.85           231              73             42   AAPL

Summary Statistics:
Best Performing Stock: AAPL (ROI: 330.55%)
Average ROI: 330.55%
Average Sharpe Ratio: 0.74
Average Maximum Drawdown: 41.85%
[*********************100%***********************]  1 of 1 completed
```

Enter Date and Thresholds



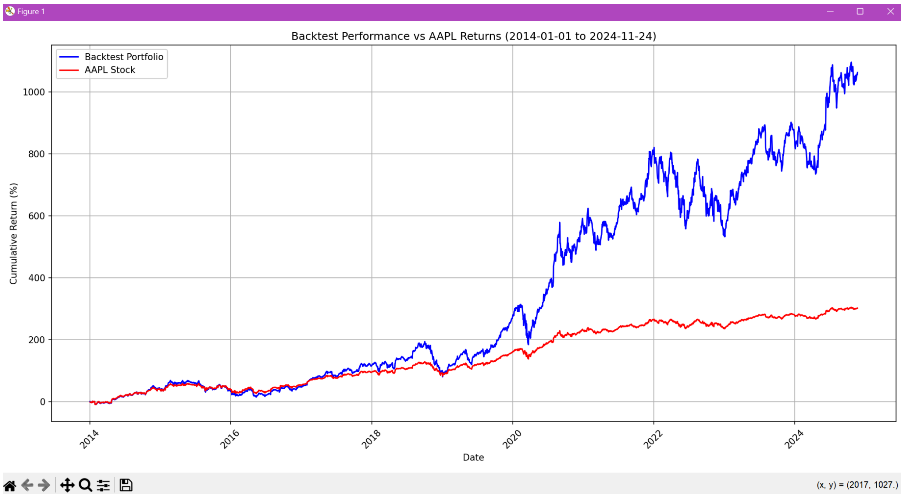
```
C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis>Python TraderBackTest.py
The stock ticker is always set to AAPL
Enter the date for sentiment analysis (YYYY-MM-DD): 2024-11-23
Enter the buy threshold (e.g., 50): 50
Enter the sell threshold (e.g., -50): -50
Looking for sentiment data at: C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalysis\
SentimentScoreDataAAPL\AAPL_2024-11-23.json
Recommendation for AAPL on 2024-11-23: hold

Would you like to run a backtest with historical sentiment scores? (y/n):
```

Run back test



```
C:\Users\geoff\OneDrive\Documents\GitHub Projects\RedditStockSentiment\SentimentAnalys
The stock ticker is always set to AAPL
Enter the date for sentiment analysis (YYYY-MM-DD): 2024-11-23
Enter the buy threshold (e.g., 50): 50
Enter the sell threshold (e.g., -50): -50
Looking for sentiment data at: C:\Users\geoff\OneDrive\Documents\GitHub Projects\Reddi
SentimentScoreDataAAPL\AAPL_2024-11-23.json
Recommendation for AAPL on 2024-11-23: hold

Would you like to run a backtest with historical sentiment scores? (y/n): y
```

Analyze Results



The trading strategy that was tested on AAPL showed strong performance, with an impressive ROI of 330.55%, indicating substantial gains over the evaluation period. This success can be attributed to the use of sentiment analysis, where posts from Reddit (specifically from r/wallstreetbets) were leveraged to generate trading signals. The Win/Loss Ratio of 1.74 suggests that for every loss, there were almost two winning trades, which reflects the strategy's effectiveness in making profitable decisions.

Despite the strong gains, the strategy also experienced a Maximum Drawdown of 41.85%, meaning there were significant periods of drawdowns where the portfolio lost nearly 42% of its value from its peak. This is a notable risk, but it is not uncommon in high-risk trading strategies, especially those influenced by market sentiment.

The Sharpe Ratio of 0.74 indicates a moderate risk-adjusted return, meaning the strategy did well in generating returns, but there is still room for improvement in terms of balancing risk. Over a total of 231 trades, there were 73 winning trades and 42 losing trades, demonstrating that the model was more successful than not, though there were some losses as expected in real-world stock trading.

In summary, while the strategy's performance was highly profitable with AAPL, it did come with significant volatility and risk, underscoring the importance of refining the model to manage drawdowns better while maintaining its profitability.

Future Works

While the current study demonstrates the potential of integrating sentiment analysis with historical stock data for trading decisions, there are several avenues for future improvements and exploration. One key area is the expansion of sentiment analysis models. While the study used VADER and FinBERT for analyzing Reddit posts, additional models such as RoBERTa or GPT-based architectures could be explored for more nuanced sentiment classification, especially in the financial domain. Furthermore, incorporating alternative sources of sentiment data, such as Twitter feeds or financial news, could provide a more comprehensive view of market sentiment and improve the robustness of the trading strategy.

Another promising direction is the integration of more advanced trading strategies. Currently, the approach relies on simple buy, sell, or hold signals based on sentiment thresholds. Future research could investigate more complex strategies that combine sentiment data with technical indicators, such as moving averages or the Relative Strength Index, or machine learning models like reinforcement learning to create dynamic trading algorithms that adapt to changing market conditions. Additionally, incorporating risk management techniques, such as stop-loss orders or portfolio diversification, could further optimize the strategy.

Backtesting with larger datasets and across multiple market conditions is also an essential step in validating the robustness of the model. Expanding the backtest to include data from various sectors and stocks, beyond just Apple, would offer valuable insights into the scalability of the approach. Furthermore, evaluating the model during different market regimes, such as bull, bear, or volatile markets, could help identify its strengths and limitations in real-world applications.

Finally, improving the real-time implementation of the trading strategy is another area of future work. The current model processes data in batches, but real-time sentiment tracking and execution of trades could be implemented to enable faster responses to market sentiment shifts. This could involve using APIs or automated trading platforms to execute trades instantly based on sentiment signals, providing an edge in high-frequency trading scenarios.

References

Wang, C., & Luo, B. (n.d.). *Predicting $GME Stock Price Movement Using Sentiment from Reddit r/wallstreetbets*. Department of Computer Science, University of Illinois at Chicago.

Python Libraries and Modules:

1. **Pandas**:
    - *pandas*. (n.d.). *pandas documentation*. Retrieved from https://pandas.pydata.org/
2. **NumPy**:
    - *NumPy*. (n.d.). *NumPy documentation*. Retrieved from https://numpy.org/
3. **yfinance**:
    - *yfinance*. (n.d.). *yfinance documentation*. Retrieved from https://pypi.org/project/yfinance/
4. **Matplotlib**:
    - *Matplotlib*. (n.d.). *Matplotlib documentation*. Retrieved from https://matplotlib.org/
5. **VADER Sentiment Analyzer**:
    - *VADER Sentiment Analyzer*. (n.d.). *VADER Sentiment Analysis library documentation*. Retrieved from https://pypi.org/project/vaderSentiment/
6. **Hugging Face Transformers**:
    - *Hugging Face Transformers*. (n.d.). *Transformers documentation*. Retrieved from https://huggingface.co/transformers/
7. **PRAW (Python Reddit API Wrapper)**:
    - *PRAW*. (n.d.). *PRAW documentation*. Retrieved from https://pypi.org/project/praw/
8. **Logging**:
    - *Python Logging Library*. (n.d.). *Logging documentation*. Retrieved from https://docs.python.org/3/library/logging.html
9. **Time**:
    - *Python Time module*. (n.d.). *Time documentation*. Retrieved from https://docs.python.org/3/library/time.html
10. **sys**:
    - *Python sys module*. (n.d.). *sys documentation*. Retrieved from https://docs.python.org/3/library/sys.html
11. **SentimentAnalyzer (Custom Module)**:
    - StockSentimentAnalyzer. (n.d.). *Custom module documentation*. Internal Documentation.