Sentiment Analysis on r/WallStreetBets and predicting the success of their Market Strategies

CSE342 - Statistical Machine Learning Course Project

Ritika Thakur
2022408
CSAI

Saksham Singh
2022434
CSE

Fig. 1. r/WallStreetBets

***Abstract*—This project aims to build a simple ML model that allows users to judge whether the market decisions in r/WallStreetBets - the biggest Reddit community for stock trading - are safe for an average person to invest in, and thus automates this process.**

I. Problem Statement and Motivation

In recent years, social media platforms such as Reddit's r/WallStreetBets have emerged as influential forums for discussions and opinions on stock investments. However, analyzing the vast amount of unstructured data generated by these communities, not to mention their colorful and profane jargon, and aggressive trading strategies, it presents a significant challenge for investors seeking to make informed decisions. The aim of this project is to develop an AI-driven solution that leverages sentiment analysis on r/WallStreetBets posts to identify potentially lucrative investment opportunities and subsequently predict the future value of these stocks.

A. Sentiment Analysis

Develop a sentiment analysis model capable of accurately assessing the sentiment expressed in posts and comments on r/WallStreetBets. The model should be trained to differentiate between positive, negative, and neutral sentiments regarding specific stocks mentioned in the forum.

B. Stock Prediction Model

Design and implement a machine learning model for predicting the future value of stocks based on historical price data, market indicators, and sentiment analysis results. The model should be capable of generating accurate predictions over various time horizons.

C. Integration and Deployment

Integrate the sentiment analysis and stock prediction models into a unified system capable of processing real-time data from r/WallStreetBets and generating investment recommendations. Deploy the system in a scalable and accessible manner, ensuring seamless operation and user interaction.

In this project we have developed the Sentiment Analysis Model and the Stock Prediction model, but not the integration of the two - which we believe is the next step in the future scope of the project.

II. Dataset Details

A. Sentiment Analysis

We have used Reddit-WallStreetBets-Posts dataset on Kaggle which features 53K posts from the subreddit r/WallStreetBets dating from 29-09-2020 to 16-08-2021. The dataset contains the post title, post body, posting timestamp, and the score (upvote-downvote sum) of the post. The data might contain a small percent of harsh language, the posts were not filtered. The dataset can be used to perform sentiment analysis, identify discussion topics and follow the trends.
For preprocessing we removed handlers, urls, special characters, single characters, substituted multiple spaces with a single space, and removed time of day from the timestamp. Then the post title and post body was divided into tokens and thus evaluated further.

B. Stock Prediction Model

Instead of using Quandl as proposed in our initial project proposal, we found out Yahoo Finance's dataset and the respective YFinance python library to be better suited for our needs but Quandl can be used too for the same with some minor changes.
Yahoo Finance hosts the data for every major listing on Nasdaq and provides the open price, the high, the low, and the close price of every single trade day for that listing since the documented history and to the present date.
We intend on using the close price variable to predict the values, thus making our model univariate. For the demonstration, we have run our model on the stock history of Apple

and Microsoft and their stock trends from 1st January 2016 to 1st January 2024. We have divided the dataset into 2 parts - namely train (65%) and test (35%).

III. Literature Review

A. Sentiment Analysis

Hutto, C.J., & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text
This seminal paper introduces the VADER lexicon, a pre-built sentiment analysis tool specifically designed for social media text. The authors propose a rule-based approach that considers not only individual words but also their context, punctuation, capitalization, and other linguistic features to accurately assess sentiment in text data.

Gilbert, E.E., & Hutto, C.J. (2014). The Structure of Online Activism
In this paper, the authors use VADER to analyze sentiment in online activist communities. They demonstrate the effectiveness of VADER in capturing nuanced sentiment expressions in social media data related to activism and social movements.

Kiritchenko, S., & Mohammad, S.M. (2018). Examining the Effectiveness of Lexical, Machine Learning, and Deep Learning Methods for Sentiment Analysis
This study compares the performance of various sentiment analysis approaches, including lexicon-based methods like VADER, machine learning algorithms, and deep learning models. The authors evaluate these methods across different datasets and domains, highlighting the strengths and weaknesses of each approach.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment Strength Detection in Short Informal Text
This paper investigates the accuracy of sentiment analysis tools, including VADER, in detecting sentiment strength in short informal texts such as tweets and Facebook status updates. The authors analyze the precision and recall of sentiment strength detection using VADER compared to other methods.

Rosenthal, S., Ritter, A., Nakov, P., & Stoyanov, V. (2015). SemEval-2015 Task 10: Sentiment Analysis in Twitter
This paper presents a sentiment analysis challenge focusing on sentiment detection in Twitter data. Participants in the challenge employed various techniques, including lexicon-based approaches like VADER, to classify sentiment in tweets. The results and insights from this challenge contribute to the understanding of sentiment analysis techniques in the context of social media data.

B. Stock Prediction Model

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions
This paper investigates the application of LSTM networks for predicting financial market movements. The authors propose a model that combines LSTM networks with technical indicators and news sentiment analysis to forecast stock prices. Their results demonstrate the effectiveness of LSTM networks in capturing complex patterns in financial time series data.

Kim, Y. (2003). Recursive neural networks for associative memory
Although not specifically focused on stock prediction, this paper introduces the concept of recursive neural networks (RNNs), which are the foundation for LSTM networks. It discusses the architecture and training of RNNs and highlights their ability to learn sequential patterns, making them suitable for time series prediction tasks such as stock forecasting.

Tsantekidis, A., Passalis, N., Tefas, A., & Kanniainen, J. (2017). Using deep learning for price direction prediction in stock market
In this study, the authors explore the use of deep learning techniques, including LSTM networks, for predicting the direction of stock price movements. They compare LSTM networks with traditional machine learning models and demonstrate superior performance of LSTM networks in capturing temporal dependencies and nonlinear patterns in financial data.

Brownlee, J. (2018). Introduction to Time Series Forecasting with Python
This book provides a practical introduction to time series forecasting techniques, including LSTM networks, using Python programming language. It covers the theoretical foundations of LSTM networks, implementation details, and practical tips for training and evaluating LSTM models on real-world datasets, making it a valuable resource for researchers and practitioners interested in stock prediction.

Zhang, Y., & Wu, Q. (2020). Forecasting stock prices with long short-term memory neural network using financial news
This paper proposes a hybrid model that combines LSTM networks with financial news sentiment analysis for stock price prediction. By incorporating textual information from financial news articles, the model enhances the predictive power of LSTM networks, enabling more accurate forecasts of stock price movements.

IV. Proposed Architecture

A. Sentiment Analysis

1) **Text Preprocessing**: This step involves cleaning the text data, including tasks like converting text to lowercase, removing special characters, URLs, single characters, and extra spaces, as well as handling stop words.
2) **Sentiment Analysis**: Utilize the VADER sentiment analysis tool, which is specifically designed for social media text. VADER provides a pre-trained model that can determine the sentiment of a piece of text, whether it's positive, negative, or neutral.
3) **Feature Engineering**: Extract relevant features from the text data that might contribute to sentiment analysis, such as the number of words, presence of specific entities (like currency mentions or organizations), average word length, etc.

4) **Data Visualization**: Visualize the sentiment distribution across the dataset, including histograms, kernel density estimation (KDE) plots, cumulative distribution function (CDF) plots, and more to understand the sentiment patterns.
5) **Temporal Analysis**: Analyze sentiment trends over time, including daily, monthly, or yearly sentiment fluctuations, using techniques like moving averages or decomposition into trend, seasonal, and residual components.
6) **Correlation Analysis**: Explore correlations between sentiment and other factors, such as post counts, to identify potential relationships or dependencies.
7) **Word Clouds**: Generate word clouds to visualize the most common words associated with positive and negative sentiment posts.
8) **Model Evaluation**: Evaluate the performance of the sentiment analysis model using appropriate metrics, such as accuracy, precision, recall, or F1-score.

B. Stock Prediction Model

We intend to use Long Short Term Memory Neural Networks (LSTM) for our architecture - majorly because of its ability to learn from sequential data while considering long-term dependencies. In the context of stock prediction, past stock prices and other relevant factors form a time series, and LSTM can effectively capture patterns and trends within this sequential data. Also their ability to handle non-linear data and memory retention, they seem to be the perfect model for our application.
We have used Tensorflow's Sequential model with LSTM and Dense layers for libraries. The model trains and predicts for each day using the trends of the last 100 days to make its decisions. We have used 3 layers for LSTM and one layer for Dense, and have computed our losses in terms of mean-squared error.

V. Results

A. Sentiment Analysis

- Due to the visible trend both in the positive and negative sentiments of the post titles, we can conclude that the data is not stationary.
- There is a general increase in positive sentiment and a general decrease in negative sentiment over time.
- The average number of posts has remained constant throughout.
- The analysis model gives a very good brief about its correlation matrices and how different factors affect the sentiment of both the post titles and bodies.
- 82% of all the discussions in the subreddit can be done with 450 words.
- As of now the model lacks the capability to classify new statements but it sure can with a little tweaking in the code

B. Stock Prediction Model

1) Apple:

- MSE on the train and test set at the last epoch of the LSTM model for each data point (mean) were 9.9508e-05 and 0.0019 respectively.
- The net MSE on train set and test set were found out to be 62.112 and 154.387 respectively.
- The future predictions seem a little flawed considering the huge dips and rises in our predictions compared to the past trends.

2) Microsoft:

- MSE on the train and test set at the last epoch of the LSTM model for each data point (mean) were 9.5480e-05 and 0.0012 respectively.
- The net MSE on train set and test set were found out to be 125.20580 and 284.346768 respectively.
- The future predictions seem a little flawed considering the huge dips and rises in our predictions compared to the past trends.

VI. Analysis of Result

A. Sentiment Analysis

- Utilizing the VADER (Valence Aware Dictionary and sEntiment Reasoner) algorithm, we conducted sentiment analysis on the titles and bodies of Reddit posts from the r/wallstreetbets subreddit. VADER proved to be effective in capturing the sentiment trends within the posts.
- The possible explanation of the frequency of words GME, robinhood around the time of January 2021 can be explained as GME's wallstreet squeeze around that time.

B. Stock Prediction Model

- LSTM proved to be a good regressor for something like stock prediction which is factored by a lot of real world variables which one might say to be random, but using the past data it was able to give a good estimate.
- Our model fails to predict for large durations in the future because of its data being referenced from another future predicted data. At best, it can give a prediction for the next day or two.

VII. References

- Inspired by Michael Reeves and his video about goldfish stock trading to make a sentiment analysis model on r/WallStreetBets
- Greg Hogg's youtube video on LSTMs and Stock prediction model - https://www.youtube.com/watch?v=CbTU92pbDKw
- Krish Naik's youtube video on LSTMs and Stock prediction model - https://www.youtube.com/watch?v=H6du_pfuznE
- Wall Street Bets Kaggle - https://www.kaggle.com/code/accountstatus/wall-streets-bet-data-analysis?rvi=1

VIII. Visualisations - Sentiment Analysis

A. Title Text Analysis

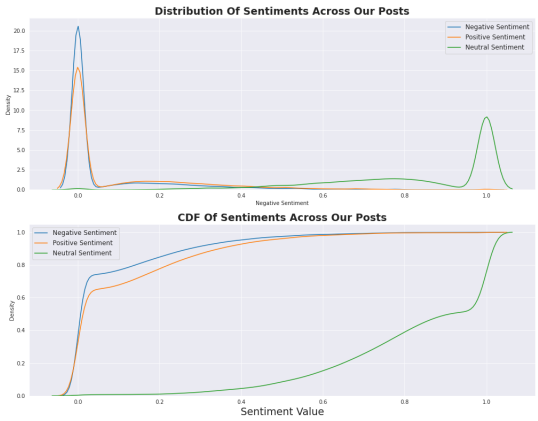

Fig. 2. Distribution of Sentiments across the posts titles, it can be observed that the majority sentiment in posts titles is neutral

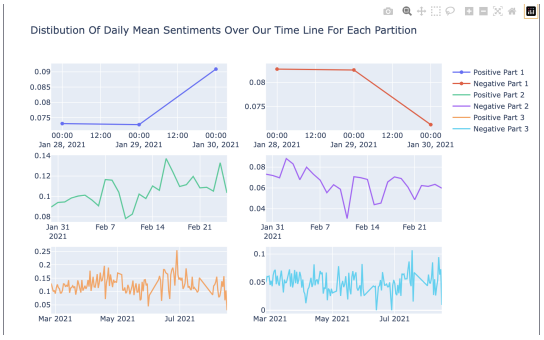

Fig. 3. Distibution Of Daily Mean Sentiments Over Our Time Line For Each Partition

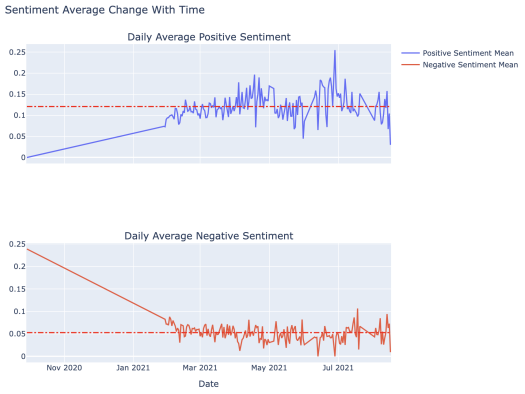

Fig. 4. Sentiment Average Change With Time

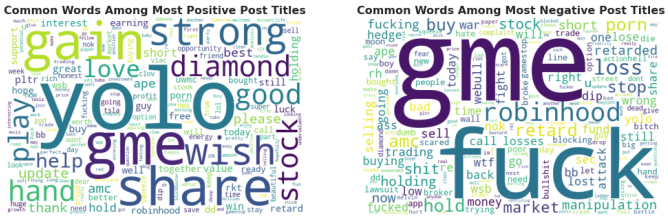

Fig. 5. Common words among the positive and negative sentiments, bigger words have higher frequency

B. Body Text Analysis

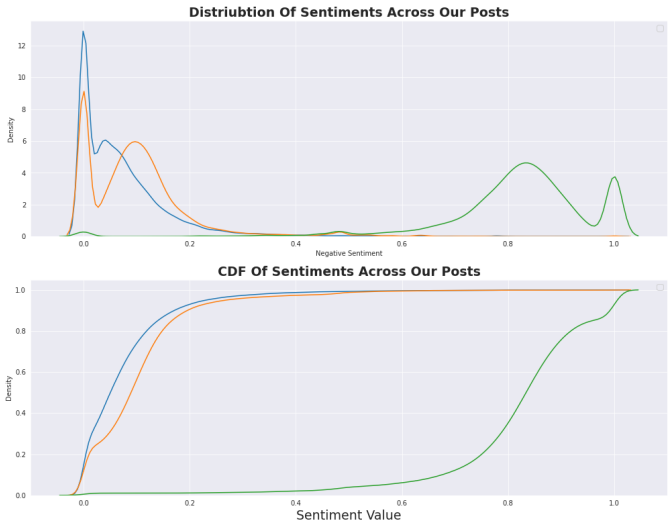

Fig. 6. Distribution of sentiments across the post bodies

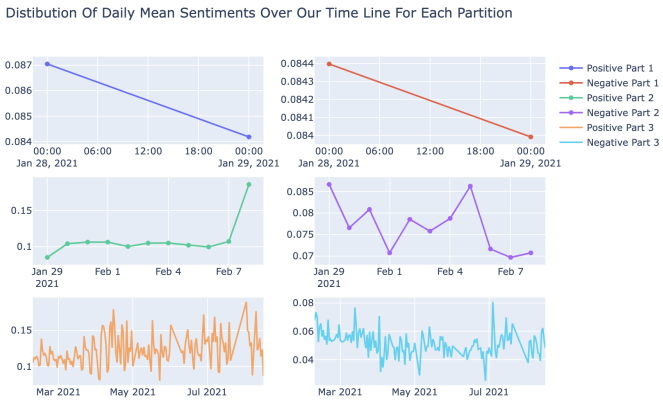

Fig. 7. Distibution Of Daily Mean Sentiments Over Our Time Line For Each Partition

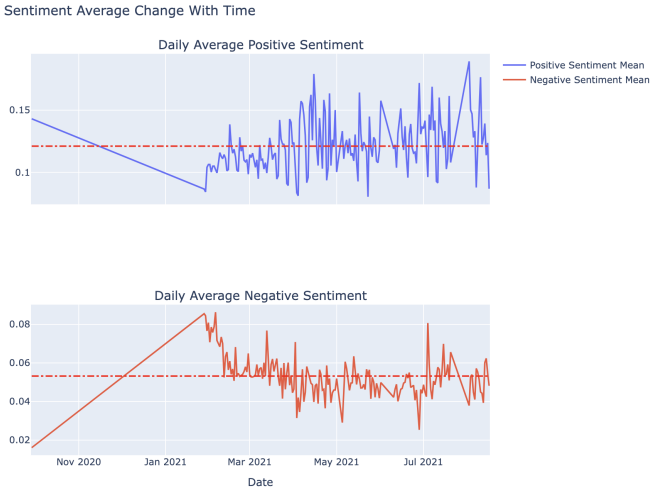

Fig. 8. Sentiment Average Change With Time

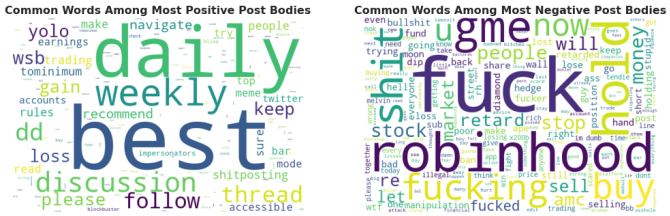

Fig. 9. Common words among the positive and negative sentiments, bigger words have higher frequency

C. Top discussions in posts

Fig. 10. Common organisations discussed in posts, bigger words have higher frequency

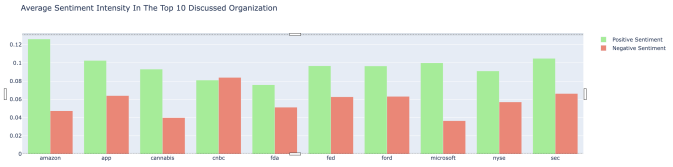

Fig. 11. Average Sentiment Intensity In The Top 10 Discussed Organization

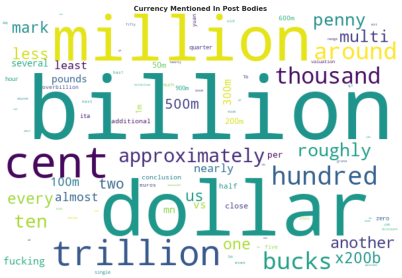

Fig. 12. Currency mentioned in post bodies

IX. Visualisations - Stock Predictor

A. Apple

The below are our findings when running the model on the Apple's stock history:

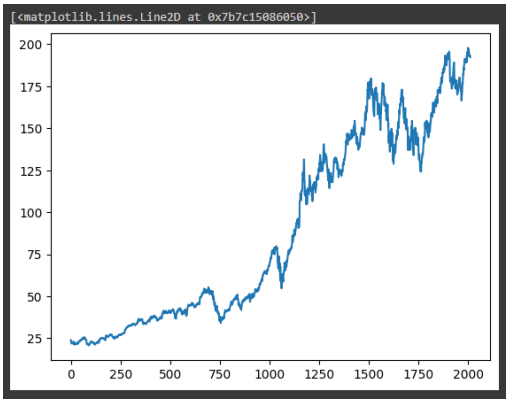

Fig. 13. Apple Dataset

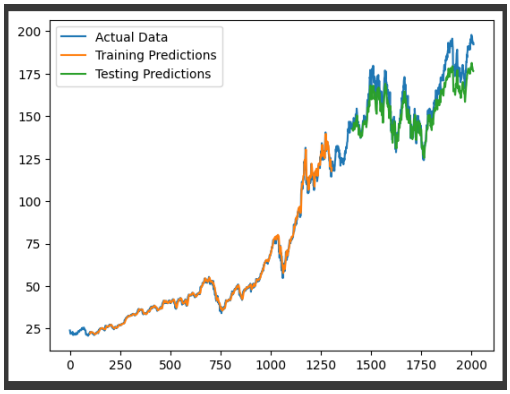

Fig. 14. Predictions on Train and Test set on the Apple Dataset

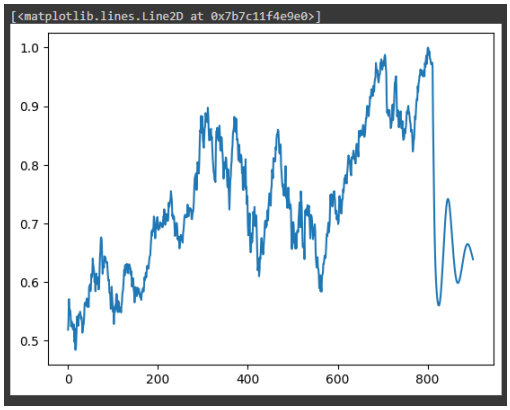

Fig. 15. Apple Predictions for next 90 days

B. *Microsoft*

The below are our findings when running the model on the Microsoft's stock history:

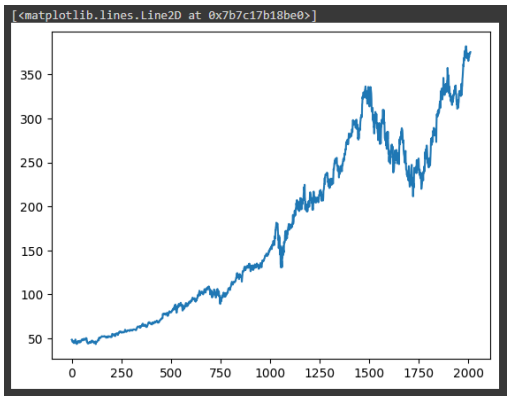

Fig. 16. Microsoft Dataset

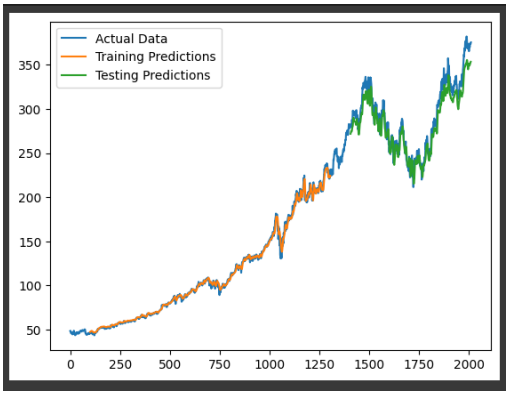

Fig. 17. Predictions on Train and Test set on the Microsoft Dataset

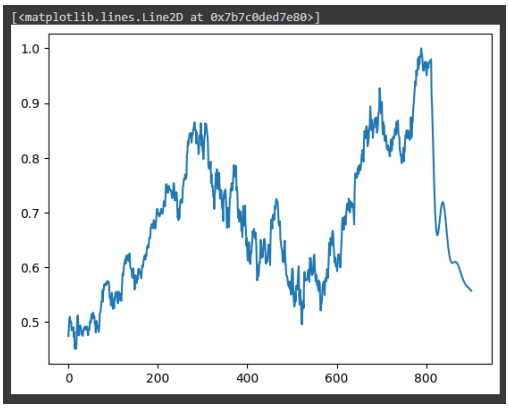

Fig. 18. Microsoft Predictions for next 90 days