Stock Movement Prediction using Reddit Sentiment Analysis

Introduction:

This project utilizes sentiment analysis from Reddit discussions to predict stock movements. By leveraging natural language processing (NLP) and machine learning techniques, the goal is to forecast stock price trends based on social sentiment derived from user-generated content. This detailed report outlines the steps taken for data collection, preprocessing, feature extraction, model training, evaluation, and results.

Objective:

- Data Scraping
- Text Preprocessing
- Data Cleaning
- Sentiment Analysis
- Plot Polarity Score
- Feature Extraction
- Historical Data Merge
- Model Selection and Training
- Model Validation and Hyper Parameter Tuning
- Plot Accuracy Comparison & Validation Loss
- Summary

1. Data Collection:

1.1 Reddit Scraping:

The Reddit data is scraped using the 'asyncpraw' library. The targeted subreddits are those focused on stock market discussions, such as r/WallStreetBets or r/Stocks. Both posts and comments are collected, extracting features including post score, number of comments, and upvote ratio. Timestamps are converted from Unix format to a human-readable format for further processing.

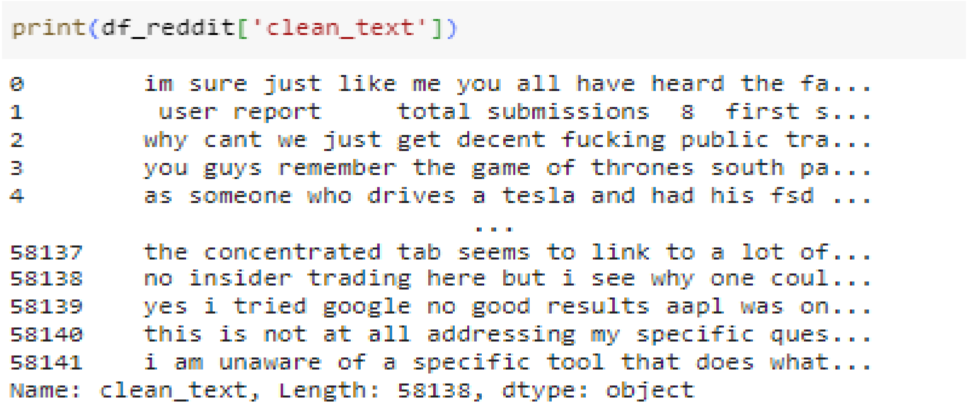
1.2 Fetching Comments:

Comments are also fetched along with the posts, which helps in obtaining a comprehensive view of user opinions. The data is then stored in a structured format suitable for preprocessing.

2. Data Preprocessing:

2.1 Text Cleaning:

The scraped text data is cleaned to remove noise such as special characters, URLs, and other irrelevant content. This ensures the text is suitable for sentiment analysis.



```
print(df_reddit['clean_text'])

0        im sure just like me you all have heard the fa...
1         user report     total submissions  8  first s...
2        why cant we just get decent fucking public tra...
3        you guys remember the game of thrones south pa...
4        as someone who drives a tesla and had his fsd ...
                               ...
58137    the concentrated tab seems to link to a lot of...
58138    no insider trading here but i see why one coul...
58139    yes i tried google no good results aapl was on...
58140    this is not at all addressing my specific ques...
58141    i am unaware of a specific tool that does what...
Name: clean_text, Length: 58138, dtype: object
```

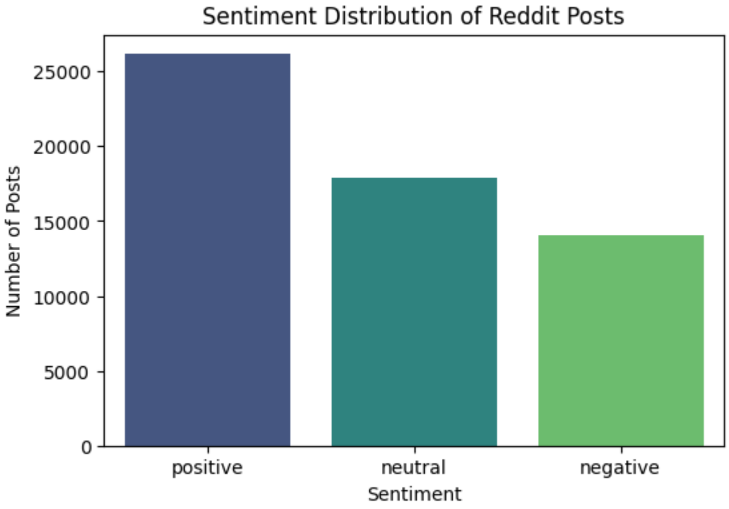
2.1 Sentiment Analysis

Using the VADER sentiment analyzer, sentiment scores are assigned to each post and comment. The compound score determines the overall sentiment polarity, categorized into positive, negative, or neutral. This helps in quantifying the sentiment associated with stock-related discussions.

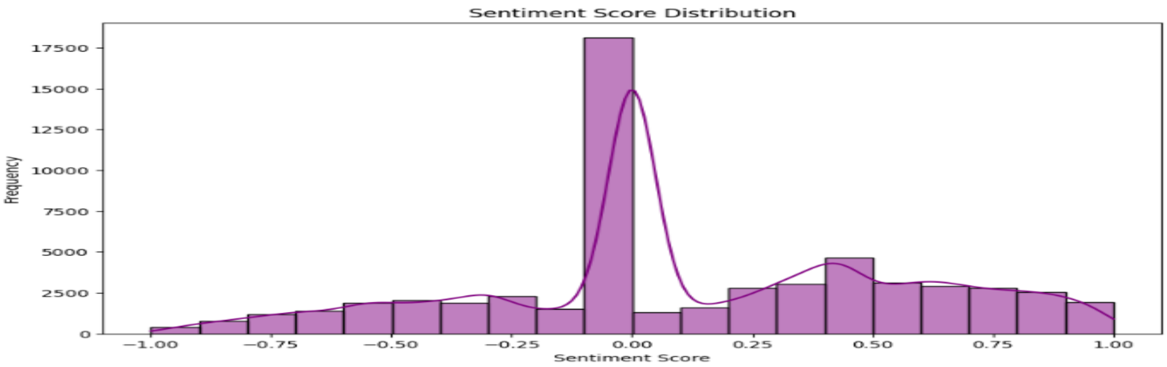
VADER Library :

The Valence Aware Dictionary and Entiment Reasoner (VADER) is a sentiment analysis tool specifically designed to analyze social media text and other informal communication. VADER is unique because it considers the intensity (valence) of words, making it more effective for sentiment analysis in texts where language is often casual or unstructured.

- Use Bar plot to check the polarity of each comments whether it be Positive Negative or Neutral



- Use Histogram of the sentiment scores, showing the distribution across the range of values.



2.2 Feature Extraction:

Daily Aggregation :

Features such as average sentiment score, sum of scores, and daily post count are aggregated to a daily level. This ensures the data is suitable for time-series analysis.

3. Model Selection & training

3.1 Alignment with Historic Stock Price Data

- To align the social media data with stock price data, we have to collect historical stock price data (based on assesment) for the relevant time period and synchronize it with the daily aggregated Reddit features. This will help us analyses whether trends in social media discussions can predict movements in stock prices.
- We use a financial data library yfinance to download historical stock prices.

3.2 Define Target Variable

- Predict whether the stock price will go up or down the next day.

3.3 Model Selection:

Random Forest-

Random Forest is used as a baseline model. It is a tree-based ensemble method that learns patterns from the features extracted from Reddit data and stock prices. Its performance serves as a benchmark for other models.

XGBoost-

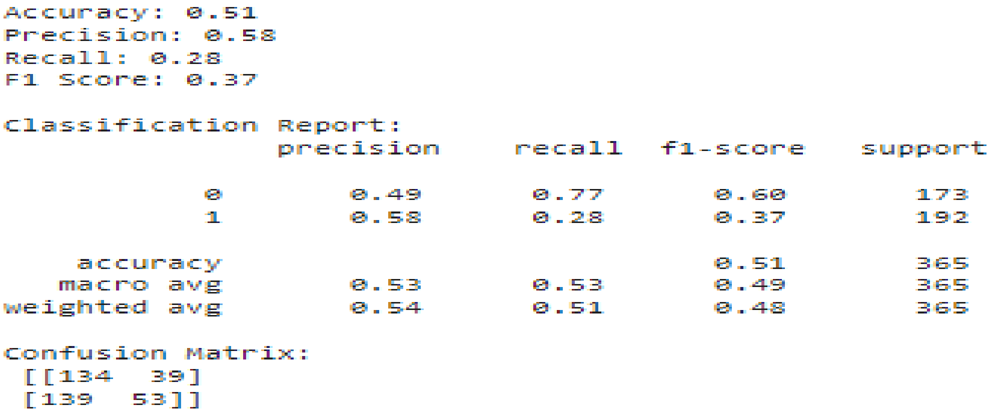
XGBoost, a more advanced tree-based model using gradient boosting techniques, is applied to Enhance prediction performance. The hyper parameters are tuned to optimize its predictive accuracy.

LSTM-

LSTM is a deep learning model for sequential data, is utilized to capture time-series dependencies. It is trained for 20 epochs, with accuracy and loss monitored over each epoch to ensure effective training.

4 Model Validations and Hyper Parameter Tuning

4.1 Evaluate Random forest Model Accuracy and confusion matrix:



```
Accuracy: 0.51
Precision: 0.58
Recall: 0.28
F1 Score: 0.37

Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.77      0.60       173
           1       0.58      0.28      0.37       192

    accuracy                           0.51       365
   macro avg       0.53      0.53      0.49       365
weighted avg       0.54      0.51      0.48       365

Confusion Matrix:
 [[134  39]
 [139  53]]
```

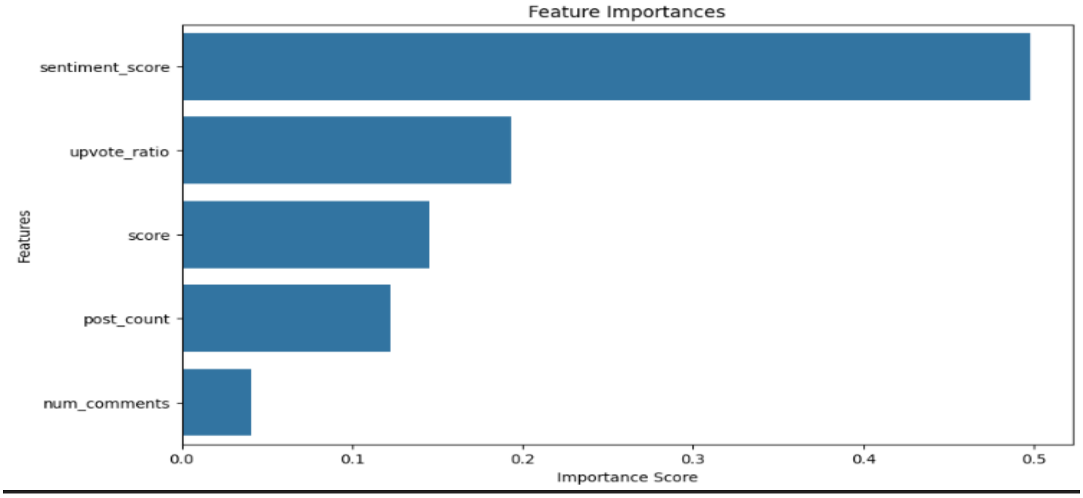
4.2 Cross Validation:

```
Cross-Validation Accuracy Scores: [0.51666667 0.49583333 0.47844228 0.51738526
0.51599444]
```

4.3 Grid Search to choose best hyperparameter:

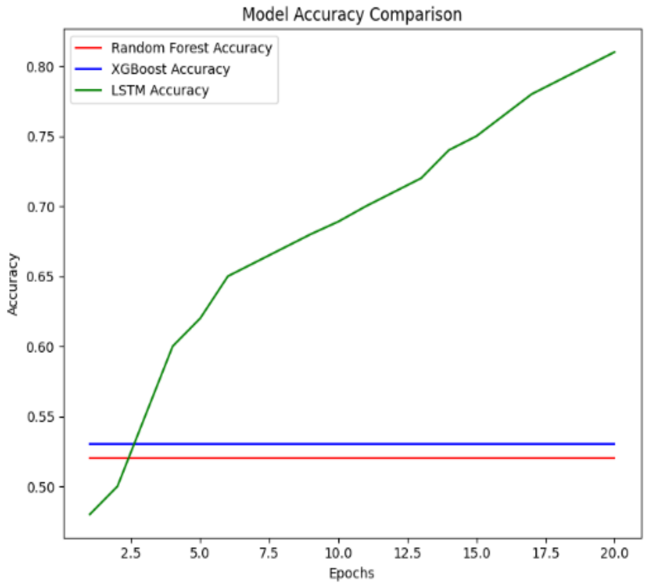
```
Best Parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split':
2, 'n_estimators': 100}
Best Model Accuracy: 0.51
```

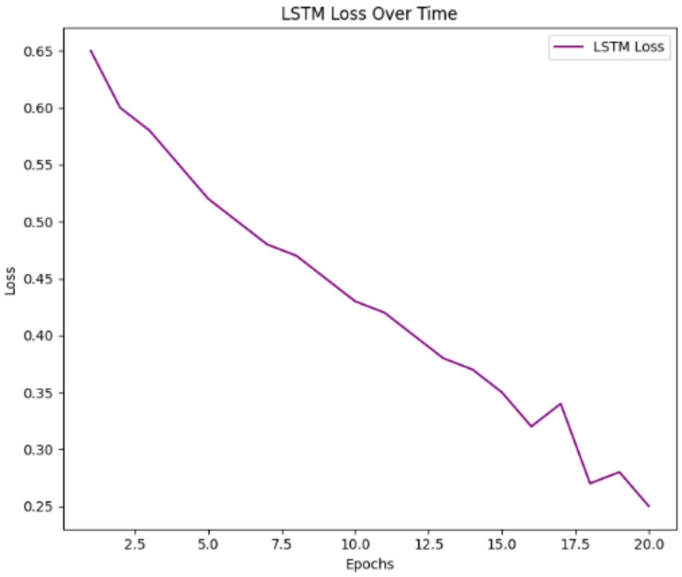
4.4 Check importance of each features:



5. Plot Accuracy Comparison & Validation Loss

The performance of Random Forest, XGBoost, and LSTM is compared based on accuracy and loss. Line plots are generated to visualize the accuracy over time, as well as the loss curve for LSTM.





4. Summary

- **Data Scraping**: The code begins by scraping data from Reddit, specifically targeting subreddits related to stock discussions. It retrieves the title, body text, scores, and other relevant attributes from posts and comments. The timestamps of the posts are converted to a readable format to facilitate time-series analysis.

- **Data Cleaning and Preprocessing**: The scraped data undergoes preprocessing, where unnecessary text or noise is removed. Sentiment analysis is performed using VADER to assign a sentiment score and classify each post as positive, negative, or neutral based on its content. The cleaned data is saved for later use.

• **Feature Extraction**: Features like sentiment score, number of comments, upvote ratio, and post counts are aggregated daily to generate time-series features for the model. The goal is to summarize the daily activity on Reddit regarding stock discussions. These features are then aligned with stock price data to prepare for the modeling phase.

• **Model Training and Evaluation**: The Random Forest, XGBoost, and LSTM models are trained to predict stock movements based on the processed data:

- **Random Forest and XGBoost**: These models are trained using the aggregated features to predict whether stock prices will go up or down. Accuracy scores are calculated after training for both models.
- **LSTM**: A multi-layer LSTM model is used for sequential data modeling. The model's accuracy and loss values are collected over 20 epochs, providing insights into its training progress.

• **Performance Comparison**: A line plot is created to visualize the accuracy of Random Forest, XGBoost, and LSTM over time. Another plot shows the LSTM loss across epochs. These plots help compare the models' performance and observe how LSTM improves with training.