Stock Market Prediction Model In Real-Time

BTECH THESIS PROJECT

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS
FOR THE AWARD OF
Bachelor of Technology
in Industrial and Systems Engineering

By

Gopireddy V N Sai Vardhan Reddy
20IM30005

Industrial and System Engineering
Bachelor of Technology (B.Tech) 2024

Under the Supervision of
Prof.Amardip Ghosh
Indian Institute of Technology Kharagpur

Department of Industrial & Systems Engineering
Indian Institute of Technology Kharagpur

Autumn 2023-24

CERTIFICATE FROM THE CANDIDATE

I wish to certify that the thesis/project report entitled, ‘Stock Market Prediction model in Real-Time submitted is completely original and has not been submitted for the award of any degree anywhere.

Yours Sincerely,
Gopireddy V N Sai Vardhan Reddy
20IM30005

(Signature with date)

CERTIFICATE FROM THE SUPERVISOR

I hereby certify that the thesis/project report entitled, "Stock Market Prediction model in Real-Time", which is being submitted by Mr. Gopireddy V N Sai Vardhan Reddy for the award of a Bachelor's degree is completely original and has not been submitted for the award of any degree anywhere. The entire work was done under my supervision and guidance.

Yours Sincerely,

Prof. Amardip Ghosh
(Supervisor)

1. Introduction:

Stock market prediction is the process of projecting a company's future developments and estimating the direction of the stock price. Shares of a corporation are exchanged on the stock market. A stock is a financial instrument that denotes ownership in a business. One location to buy those equities is the stock market. Acquiring stock in a corporation entails acquiring a minor stake in an establishment. In order to create a model that accurately predicts the stock price based on current market patterns, we are forecasting stock prices using a machine learning algorithm. I have made accurate stock price predictions using LSTM recurrent neural networks. There are two types of stocks that we would find. In the first, we would primarily focus on buying and holding the stock, as this is how most traders operate because they often experience large short-term losses. This means that a stock is bought at a price and then sold when the trader's level of satisfaction is reached on a given trade.

LSTMs are very important, as they are very powerful in sequence prediction problems because they can store previous or past information. This is very important in stock prediction as we need to store and read the previous stock information as well to forecast the stock prices accurately in the future.

2. Data Collection:

For the experimental study, we downloaded live datasets namely google, or nifty reliance, etc can take any type of stock ticker. from the Yahoo Finance website (https://finance.yahoo.com/).

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-01-02	29.667933	30.151802	29.620493	29.724857	20.950949	16371571
1	2015-01-05	29.743834	29.800758	29.421251	29.563566	20.837263	24786391
2	2015-01-06	29.667933	30.227703	29.525618	29.810247	21.011133	29468681
3	2015-01-07	30.094877	30.237192	29.962049	30.218216	21.298689	20248816
4	2015-01-08	30.683111	30.967743	30.569260	30.834915	21.733349	49169522

Explanation of the above code is as follows Import yfinance as yf: The script starts by importing the yfinance library as yf. yfinance is a popular Python library that allows users to download historical market data from Yahoo Finance, which can be useful for financial analysis, stock market predictions, or just keeping track of stock performance.
Function Definition - download_stock_data(symbol, start_date, end_date): Defines a function named download_stock_data that takes three parameters:

- symbol: The stock's ticker symbol (e.g., "AAPL" for Apple Inc.).
- start_date: The starting date of the period for which the data is to be downloaded, in "YYYY-MM-DD" format.
- end_date: The ending date of the period for which the data is to be downloaded, also in "YYYY-MM-DD" format.

Try-Except Block: Inside the function, a try-except block is used to handle any potential errors that might occur during the data download process.

Function Call: The download_stock_data function is called with the user-provided symbol, start_date, and end_date as arguments. The script attempts to download the data for the given parameters and save it as a CSV file.

<u>3. Model Architecture:</u>

Predicting stock market prices involves understanding and analyzing complex patterns in historical data. While the LSTM model is a popular choice for its ability to capture temporal dependencies, various other algorithms and techniques can be used for stock market prediction, each with its unique strengths and applicabilities. Here's an overview of some common types:

1. Linear Regression
Description: A simple method that uses a linear equation to represent the link between a dependent variable (stock price, for example) and one or more independent variables (trading volume, historical prices, etc.).
Ideal for: Fast analysis, basic model development, and simple patterns and correlations in data.
2. ARIMA (Autoregressive Integrated Moving Average)
This is a traditional time series forecasting technique that incorporates a moving average (MA) model, differencing (I) to keep the time series stationary, and autoregression (AR).

Best for Time series data that is primarily univariate, where the future value has a linear relation with past values and errors.

3. Random Forest

The danger of overfitting associated with single decision trees is decreased by employing many decision trees in an ensemble learning technique.

Ideal for Feature importance analysis, capturing nonlinear connections in data, and handling mixed numerical and categorical data.

4. Support Vector Machines (SVM)

Description: Primarily used for classification tasks but can be adapted for regression (SVR - Support Vector Regression). SVMs find the hyperplane that best separates different classes (or predicts values) in the feature space.

Best for High-dimensional data, capturing complex relationships, and when the data has a clear margin of separation.

5. Gradient Boosting and XGBoost

Gradient Boosting is an ensemble strategy that develops models in a sequential fashion, with each new model fixing the mistakes of its predecessors. XGBoost is a distributed gradient boosting library that has been built for maximum efficiency, versatility, and portability.

Ideal for: Managing diverse data kinds, resilience to anomalies in the output domain, and scenarios where precision in forecasting is crucial.

6. Neural Networks and Deep Learning

Convolutional Neural Networks (CNN): By treating sequences as one-dimensional pictures, CNNs may be modified to handle time series data, despite their primary application in image processing.

Because recurrent neural networks (RNNs) can retain state or memory across sequences, they are perfect for sequential data, such as market prices.

LSTM (long short-term memory) and GRU (gated recurrent unit): RNNs that have been upgraded to capture long-term dependencies without experiencing the vanishing gradient issue.

Transformer Models: Because they can handle sequential data without requiring sequential processing, transformers—which were first created for natural language processing tasks—have showed promise in time series forecasting.

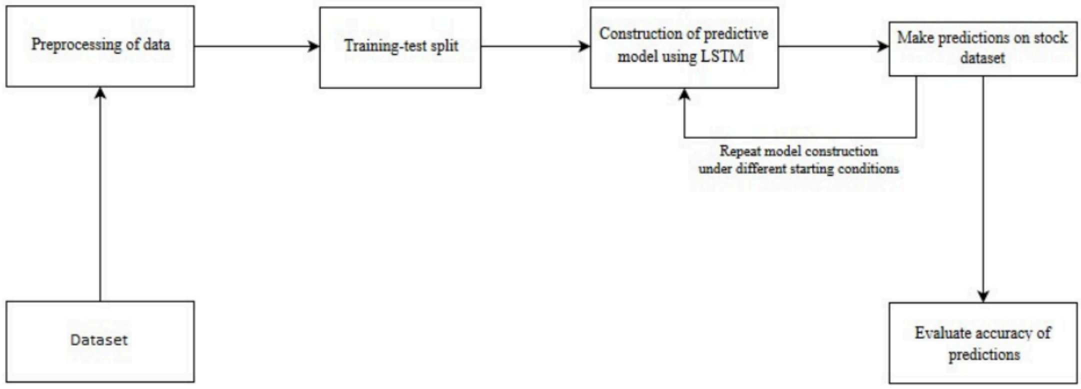
Ideal for: Capturing temporal dependencies, intricate patterns and interactions in huge datasets, and situations in which you have access to a lot of processing power.

7. Reinforcement Learning

Models learn policies that maximize a cumulative reward in order to be educated to produce decision sequences. employed especially in algorithmic trading techniques, in which the model is trained to make trades in order to maximize profit.

Ideal for: Creating trading strategies where choices are made in a sequential manner depending on the results of earlier activities.

Long Short-Term Memory



The Above diagram illustrates a process flow for stock market prediction using a machine learning model, specifically an LSTM (Long Short-Term Memory) network. It begins with a dataset that is first subjected to preprocessing, a step critical for cleaning and preparing the data for analysis, often involving normalization, handling missing values, and feature selection. Following this, the data is divided into training and test sets, which is essential for training the model on one subset of the data and validating its performance on another, unseen subset.

Next, the preprocessed training data is used to construct a predictive model based on the LSTM architecture, a type of recurrent neural network well-suited for sequential data such as stock prices. This LSTM model is then used to make predictions on the stock dataset. It's common practice to iterate over the model construction phase multiple times with different starting conditions to refine the model and avoid local minima in the learning process.

Finally, the model's predictions are evaluated against actual outcomes to determine the accuracy of the model. This step is crucial for assessing the performance of the model and can involve a variety of metrics such as mean squared error, mean absolute error, or other domain-specific accuracy measures. The evaluation helps in understanding how well the model is likely to perform when making predictions on real stock market data.

4)Prediction:

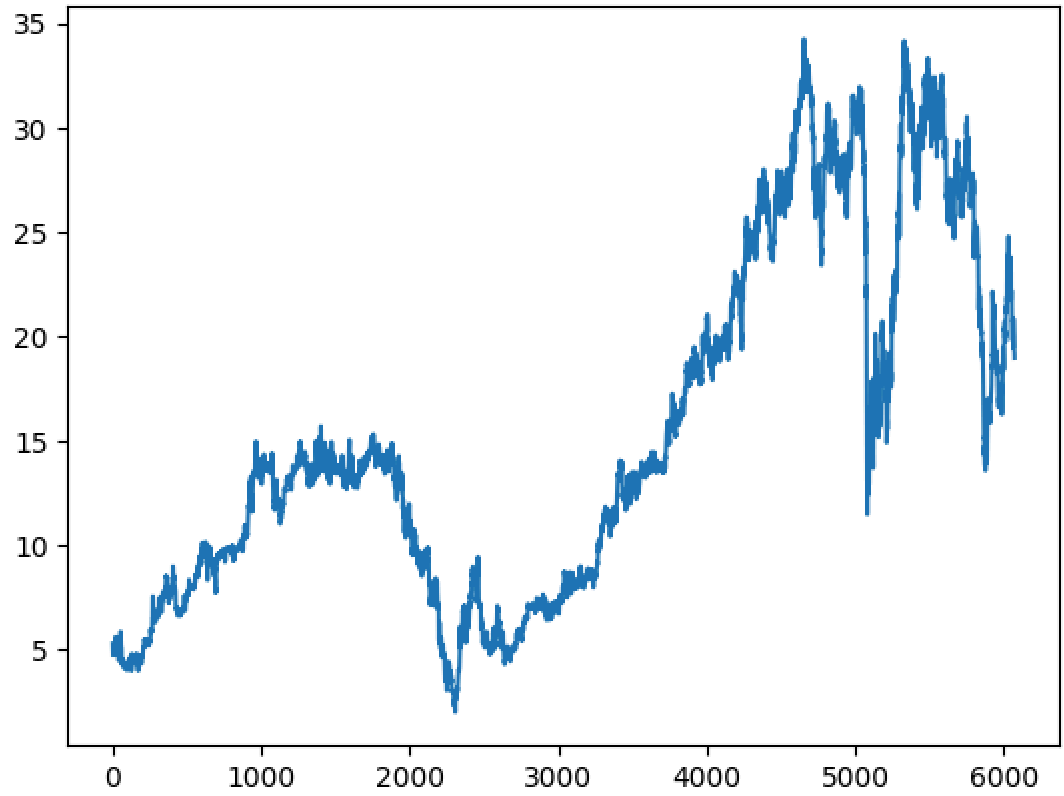
Now let us take into consideration the stock **GOOG(Google)**. conditions taken into consideration are

```
Enter the stock symbol: GOOG
Enter the start date (YYYY-MM-DD): 2000-01-01
Enter the end date (YYYY-MM-DD): 2024-03-29
[*********************100%%**********************]  1 of 1 completedData downloaded successfully!
```

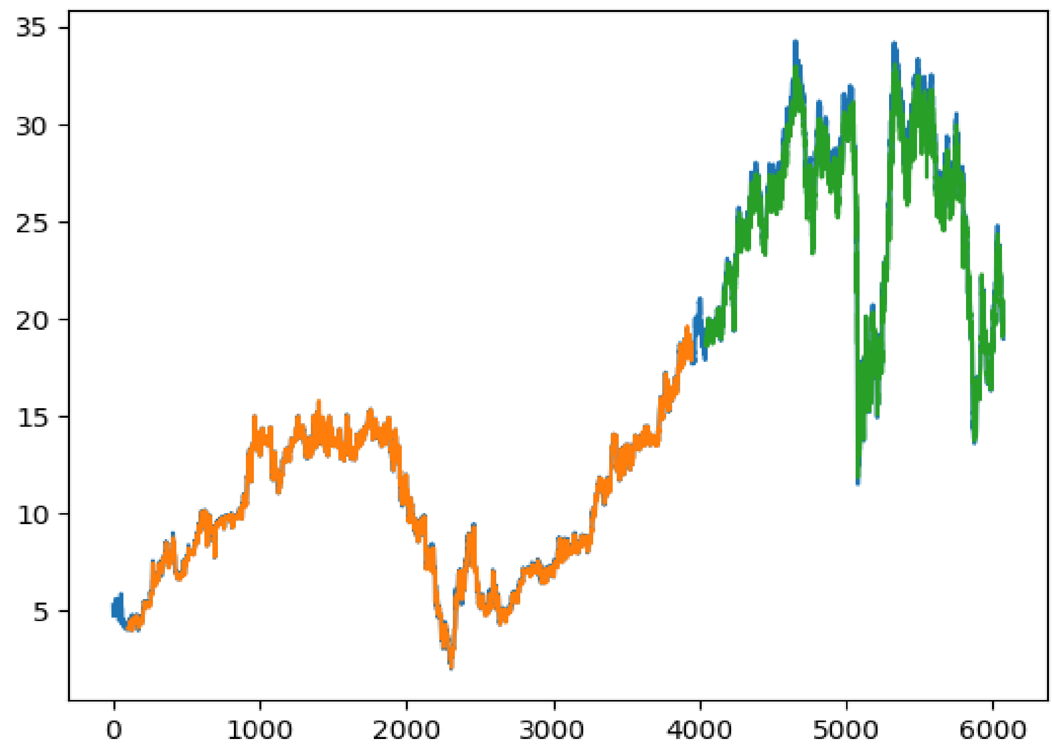
Data collection (sample):

Date	Open	High	Low	Close	Adj Close	Volume
2000-01-03	30.419828	30.716318	30.004744	30.241936	13.061069	12873345
2000-01-04	29.648956	29.886148	28.462997	29.115274	12.574474	14208974
2000-01-05	29.293169	30.241936	29.233871	29.589659	12.779362	12981591
2000-01-06	29.648956	31.072105	29.589659	30.657021	13.240336	11115273
2000-01-07	32.258064	33.088234	30.657021	32.732449	14.136687	17962163

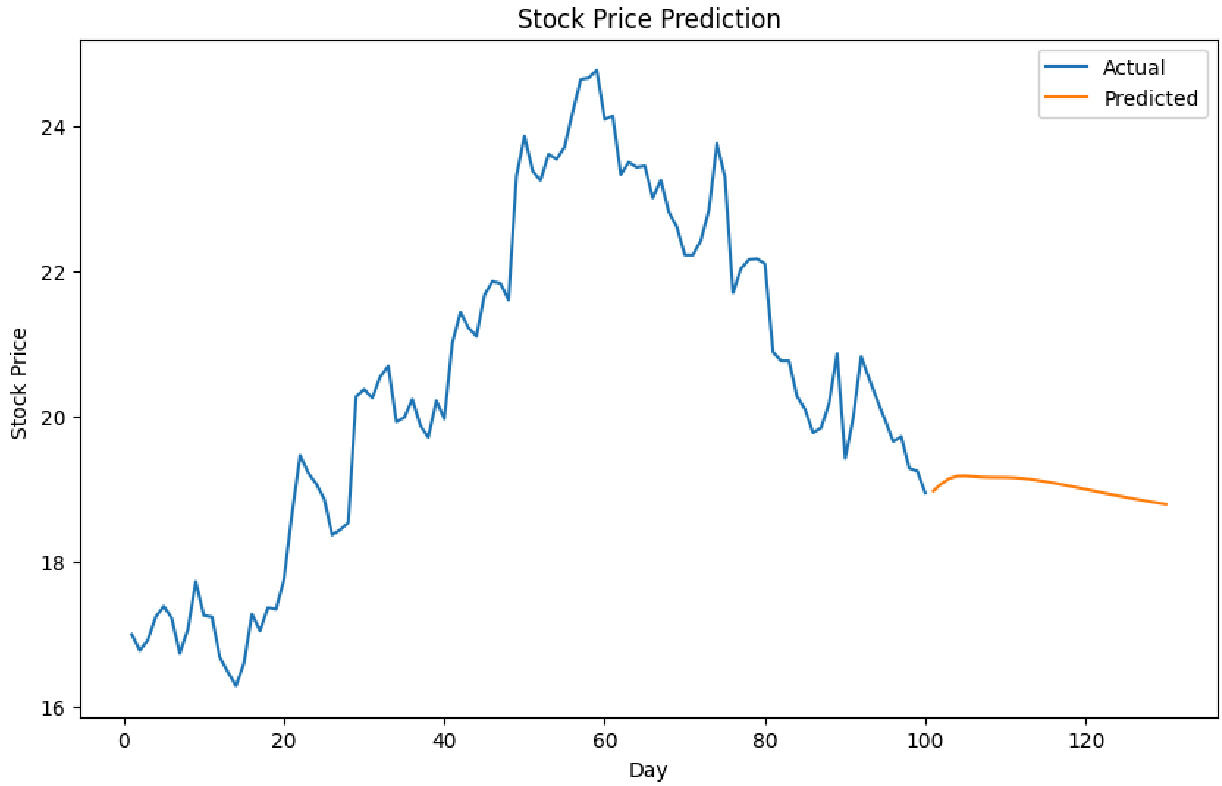
Actual data:



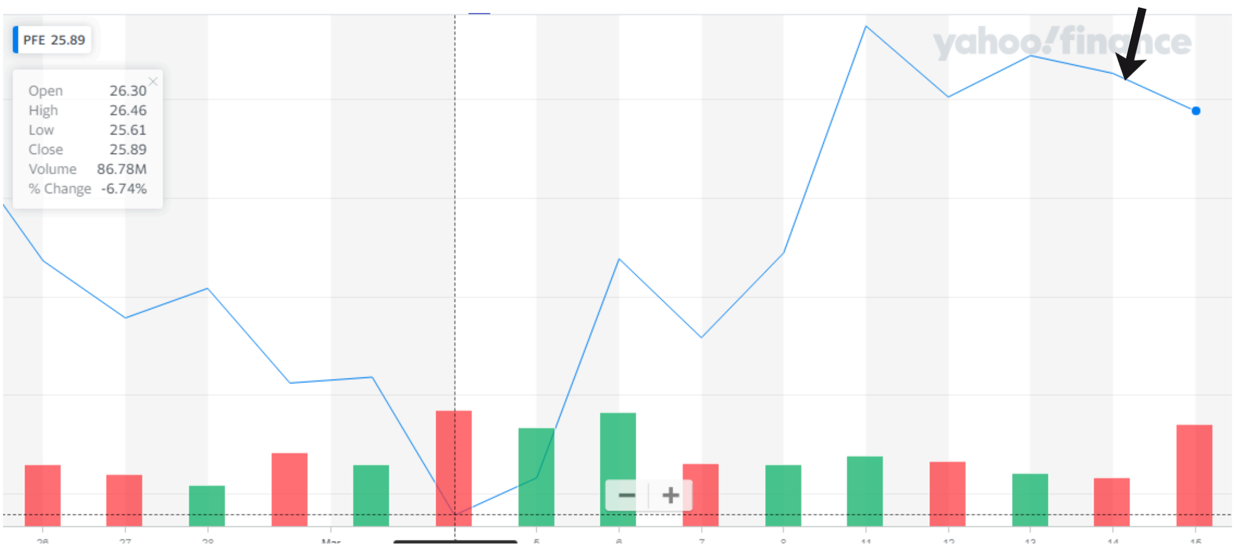
training and testing the data:



orange line - train predicted || green is test predicted



Actual stock :



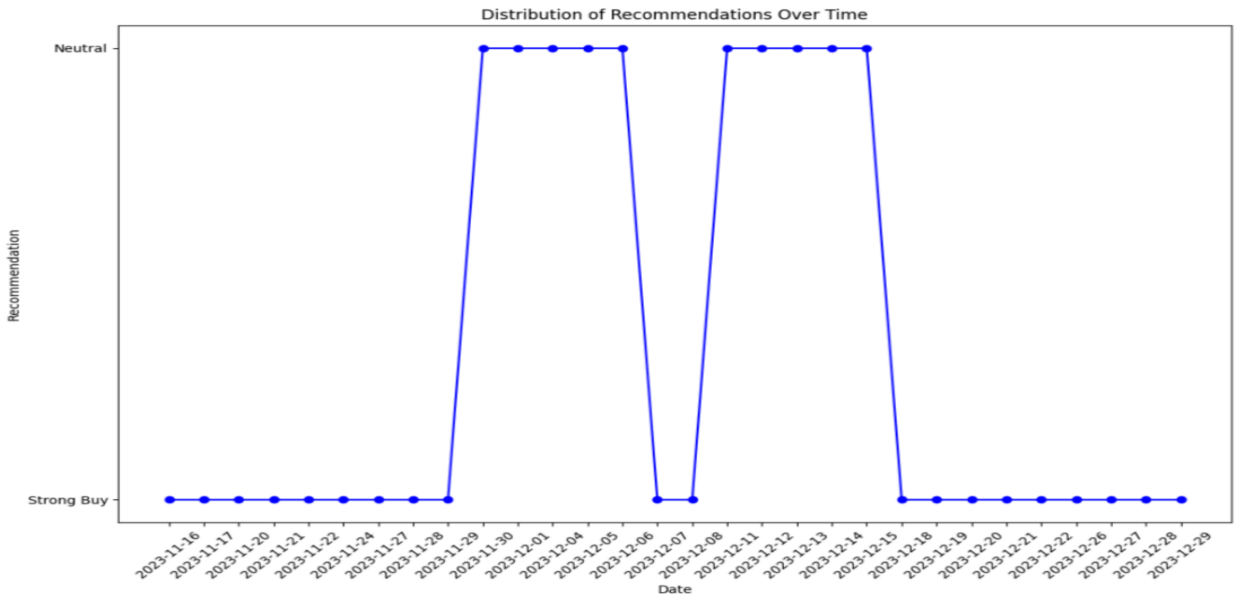
5. Recommendation to buy or hold the stock:

The utilization of the Simple Moving Average (SMA) and Exponential Moving Average (EMA) as technical analysis tools to provide trading signals is the main emphasis of this research. Based on past data, these indicators are frequently employed in the financial markets to forecast future price fluctuations.

Compute Indicators:
1) Simple Moving Average (SMA): The rolling mean approach is used to determine the 20-day SMA. This functions as a trend-following indicator for the stock price.
2) Exponential Moving Average (EMA): The exponential weight algorithm used to generate the 20-day EMA gives greater weight to more recent data, making it more sensitive to fresh information.

Signal Processing:
1)Signal Generation: Trading signals are generated based on the position of the 'Close' price relative to the SMA and EMA. A signal is marked as 1 if the 'Close' price is above the respective average and 0 otherwise.
2)Signal Aggregation and Interpretation: The signals from SMA and EMA are aggregated. A total signal score of 2 indicates a strong buy recommendation, while a score of 0 or 1 leads to a neutral position.

Recommendation Logic:
1)The system generates a 'Strong Buy' recommendation if both SMA and EMA signals agree on a bullish trend (total signal of 2).
2)The recommendation defaults to 'Neutral' for other scores, indicating mixed or lack of clear signals from the indicators.



6. Web-Scrapping Symbol News from Yahoo:

This is designed to automate the process of scraping news articles from Yahoo News search results. This script uses various Python libraries to request, parse, and write news data to a CSV file. This allows for the efficient gathering of news articles based on specific search terms, which can be utilized for data analysis, research, or monitoring purposes.

Important Libraries Used:

- re: Offers regular expression matching functions like to those in Perl, which are utilized in this application to extract particular segments of URLs.
- csv: Provides classes for the reading and writing of tabular data in CSV format.
- time: To prevent flooding the server or becoming blocked, requests are specifically delayed using the sleep function.
- BeautifulSoup: A library that offers Pythonic idioms for iterating, finding, and altering the parse tree, making it simple to scrape data from websites.
- requests: Enables Python-based HTTP request transmission, facilitating communication with online services.

Functional Description

Headers Setup: A dictionary of HTTP headers is prepared to simulate requests from a web browser, which helps bypass certain kinds of basic bot protections.

get_article Function:
Input: A card element parsed by BeautifulSoup representing an individual news article.
Process: Extract relevant details such as headline, source, time posted, description, and the article link. Uses regular expressions to clean the URLs extracted from the card elements.

Output: Returns a tuple containing the cleaned data of the article.
get_the_news Function:

Input: A search string.

Process:
Constructs a search URL and fetches the page content.

Iteratively parses the HTML content to extract news cards and uses get_article to process each card. Keeps track of unique links to avoid duplicates.

Handles pagination by finding and requesting the link to the next page until no more pages are found.

Output: Saves all the articles into a CSV file and also returns a list of article tuples.

Implementation Details

- Data Extraction: The script effectively navigates through the structure of Yahoo's news search result pages, extracting structured data from what is inherently unstructured HTML.
- Error Handling: Basic error handling is implemented for pagination, using try-except blocks to determine when no further pages are available.

- Throttling Requests: Includes a deliberate delay (sleep) between requests to respect the server's load and minimize the risk of being blocked for too frequent requests.

```
('Google is dragging its feet on killing third-party cookies yet again',
 'Android Central',
 '2 days ago',
 "Google's plan to eliminate third-party cookie trackers has been postponed until next year. The UK's Competition and Markets Authority (CMA) has raised concerns about",
 'https://www.androidcentral.com/apps-software/google-is-dragging-its-feet-on-killing-third-party-cookies-yet-again'),
('Google is updating Android TVs to fix a big Gmail privacy problem',
 'The Verge',
 '2 days ago',
 'Google is working on a fix to block crafty individuals from gaining access to the emails of accounts logged in to some Android TV units. The Android TV operating system, which ...',
 'https://www.theverge.com/2024/4/25/24140152/google-android-tv-gmail-email-hack-browser-expose'),
('Google parent reports another quarter of accelerating growth and its stock price soars',
 'Quartz',
 '2 days ago',
 'Google’s corporate parent Alphabet Inc. on Thursday released a quarterly report showing it’s still...',
 'https://qz.com/google-parent-reports-another-quarter-of-accelerating-g-1851436521'),
('Google parent reports another quarter of robust growth, rolls out first-ever quarterly dividend',
 'WSB Radio',
 '2 days ago',
 'Google’s corporate parent Alphabet Inc. on Thursday released a quarterly report showing it’s still...',
 'https://www.wsbradio.com/news/google-parent/DALCZQPI5FUAPJYIYFRJSHVGPA/'),
('Google Introduces New AI Training Course',
 'Forbes',
 '2 days ago',
 'Google has announced a new major training initiative designed to increase AI skills throughout the workforce. The project includes a new Google AI...',
 'https://www.forbes.com/sites/michaeltnietzel/2024/04/26/google-introduces-new-ai-training-course/'),
```

6. Sentiment Analysis:

- **Python Script Segment Overview**

The Python script segment employs the TextBlob library to perform sentiment analysis on news headlines. It then uses this sentiment to make simple financial recommendations (Strong Buy, Sell, Hold) based on the sentiment scores of the headlines.

- **Libraries and Tools**

TextBlob: A Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, and sentiment analysis.

- **Functional Description**

Sentiment Analysis:

Input: The headlines of news articles.

Process: Each headline is passed to TextBlob, which computes a sentiment polarity score. This score is a float within the range [-1.0, 1.0], where 1.0 denotes positive sentiment and -1.0 denotes negative sentiment.

Output: A sentiment score for each headline.

Financial Recommendation Generation:

Input: Sentiment scores from the analysis.

Decision Criteria:
A score of 0.7 or higher triggers a 'Strong Buy' recommendation.
A score of -0.7 or lower results in a 'Sell' recommendation.

Scores between these thresholds receive a 'Hold' recommendation.

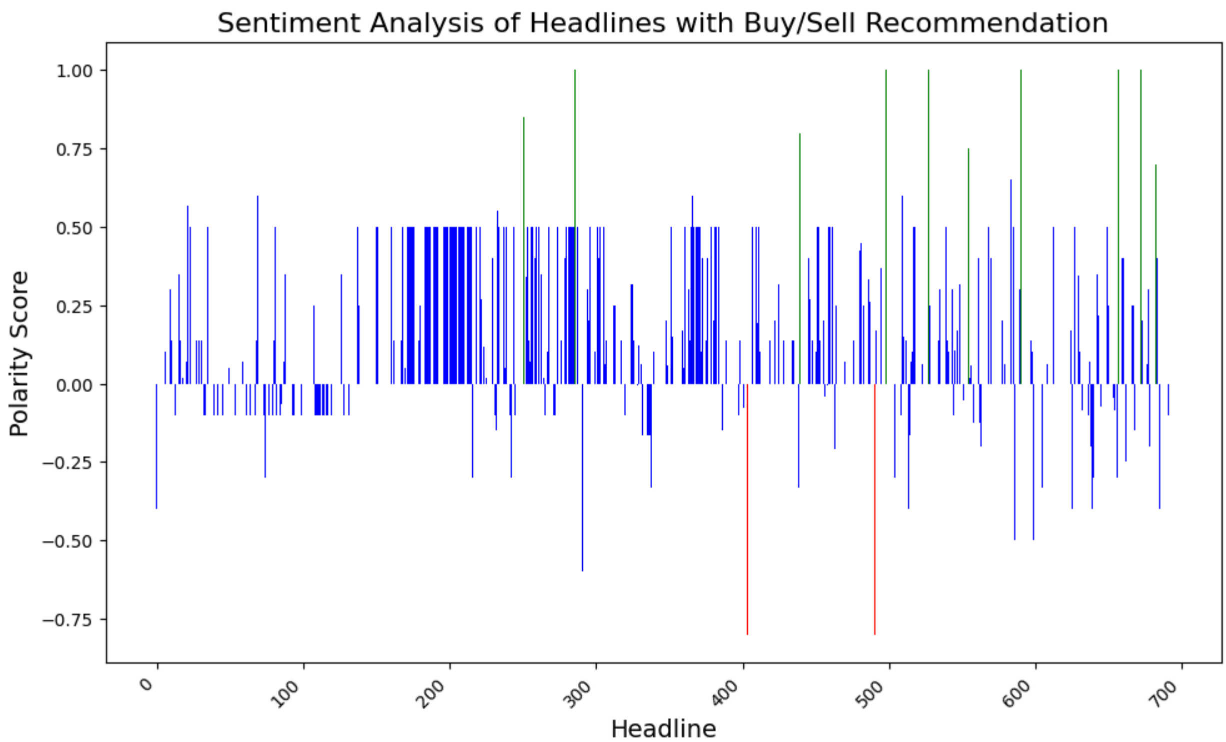
Output: A list of tuples, each containing the article details and the corresponding financial recommendation.

- **Error Handling and Limitations**

The script does not handle neutral sentiment distinctively unless it falls exactly within the -0.7 to 0.7 range, where it defaults to 'Hold'.
The sentiment analysis is purely based on the headlines and does not consider the context or content of the articles, which can lead to misleading recommendations if the headline is not reflective of the article content.

Usage

This script is particularly useful for financial analysts and investors who want to quickly gauge market sentiment regarding certain stocks, sectors, or events based on news headlines. However, users should be cautious and consider additional analysis or sources before making investment decisions, as sentiment analysis alone does not provide a comprehensive view of market conditions.



This image explains to us which news is mostly important and that could affect the valuation of the stock which is further analyzed to say whether to buy a stock keep the position in a hold or sell the stock

7. Conclusion:

This research project integrates advanced predictive modeling and sentiment analysis to provide actionable insights for stock market investments. By utilizing Long Short-Term Memory (LSTM) neural networks, the model leverages historical stock price data to forecast future price movements over a 30-day horizon. This approach captures the inherent trends and patterns in the stock price sequences, allowing for a sophisticated analysis of potential future values based on learned dependencies.

In parallel, the project employs a sentiment analysis model that scrutinizes news headlines from Yahoo Finance. By analyzing the sentiment conveyed in these headlines, the model assesses the public sentiment surrounding specific stocks or the market in general. This sentiment analysis is crucial as it encapsulates the market's psychological atmosphere which can significantly influence stock movements.

The combination of LSTM-based predictive analytics and sentiment-driven recommendations provides a dual-layered approach to stock market forecasting. For each stock analyzed, the model not only predicts future price movements but also evaluates current news sentiment to offer recommendations—Strong Buy, Sell, or Hold. This methodology acknowledges that stock prices are influenced by both historical trading patterns and the prevailing market sentiment, often driven by recent news and events.

The strengths of this model lie in its holistic approach to stock market analysis, blending quantitative data with qualitative insights. However, potential limitations include the reliance on the accuracy and relevance of the input data, such as the quality of the news headlines used for sentiment analysis and the assumption that past price patterns will persist into the future.

Overall, this model represents a significant step forward in the use of artificial intelligence for financial decision-making. It provides a robust tool for investors who seek to harness the power of machine learning and natural language processing to enhance their investment strategies. Future enhancements might include the integration of real-time data feeds and the application of more nuanced sentiment analysis techniques to further refine the accuracy and reliability of the predictions and recommendations.

References:

[1] Soumik Bose1, GiridharMaji, Narayan C. Debnath, SoumyaSen, Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh.

[2] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman- Stock price prediction using LSTM, RNN, and CNN-sliding window model-2017

[3]Stock Price Prediction Using News Sentiment Analysis Saloni Mohan; Sahitya Mullapudi; Sudheer Sammeta;Parag Vijayvergia; David C. Anastasiu

[4]**Using Market News Sentiment Analysis for Stock Market Prediction by Marian Pompiliu Cristescu 1ORCID,Raluca Andreea Nerisanu,Dumitru Alexandru Mara**