**AAPL Stock Forecasting based on lstm model and sentiment analysis**

Data Acquisition and Processing Systems ELEC0136 23/24 report

*SN: 23202440*

#### Abstract

This research explores stock analysis and forecasting for Apple Inc. (AAPL) using historical adjusted close prices and sentiment analysis from 2019 to 2023. Data was acquired through APIs, including IEX Cloud for stock prices and New York Times and Financial Times for news headlines. The study integrates visualization tools, ARIMA models, and LSTM neural networks for data exploration, time series forecasting, demonstrating improved predictions when sentiment analysis is incorporated. Decision-making strategies are developed based on predicted values and return thresholds. The study highlights the importance of comprehensive data analysis and forecasting for effective stock market decision-making.

**Index Terms—** Stock Price Prediction, Sentiment Analysis, LSTM, ARIMA, Time Series Forecasting

**1. Introduction**

The abundance of techniques stemming from machine learning and deep learning fields has significantly bolstered stock prediction capabilities. The aim of this project is to analyze the market trend of a specific company and provide recommendations for buy, hold, or sell actions. My research focused on Apple Inc Stock (AAPL), and the whole system can be adapted for application to other companies across various time periods.

I obtained stock data from IEX Cloud through an API, covering the period from April 1, 2019, to April 30, 2023. Additionally, I collected news headlines related to AAPL using APIs from both the New York Times and Financial Times within the same timeframe. Subsequently, sentiment intensity analysis was conducted using the NLTK package. The resulting sentiment-analyzed news data was integrated with the stock data and stored in MongoDB Atlas. To facilitate the management of data, I created a REST API server with Flask, allowing for the implementation of CRUD functions through the database.

Following the connection to MongoDB Atlas, stock and sentiment data are obtained and preprocessed for further development. Through Exploratory Data Analysis (EDA), I analyze the variation trend, seasonality, news-market correlations, on-balance volume indicators, and forecasting confidential intervals, utilizing calculations, visualization tools, and the ARIMA model. For time series forecasting, I employed the LSTM model, adjusting window size, historical period, and sequential layers to achieve a well-matched model. A comparison between models with and without sentiment signals, using RMSE measurement, demonstrates the valuable impact of sentiment analysis. The integration of financial indicators and forecasting proves effectiveness for decision-making.

The report is structured into distinct sections, starting with a detailed data description that covers sources, formats, content, etc. Following this, the acquisition approaches through different APIs with initial processing are introduced. The subsequent sections explain the data storage method and REST API implementation. Further sections delve into preprocessing methods, Exploratory Data Analysis (EDA) on the data, deep learning-based forecasting, and decision-making. The next section presents the experimental results, provides parameter selection and comparison between different parameters, and displays the predicted results. Finally, the report provides a comprehensive summary, providing insights into the Apple stock market and directions for future technical improvements.

**2. DATA DESCRIPTION**

In this project, two types of data were collected for stock forecasting: historical stock price data from 2019-04-01 to 2023-04-30 and corresponding news headlines from the same period.

The primary dataset for this project is the Apple Inc stock (AAPL) data, acquired in raw format through the IEX Cloud API in .json format. The selected data columns include Price Date, fhigh, flow, fopen, fclose, and volume, where the 'f' prefix indicates full adjustment for historical dates.

Financial news is regarded as a key market information source. I gathered data from New York Times and Financial Times. Corresponding API requests and BeautifulSoup were utilized for data extraction, integrating filter words to scrape Apple-related news headlines for sentiment analysis. The resultant datasets are stored in lists containing the published date and full headlines obtained through scraping.

**3. DATA ACQUISITION**

Data acquisition marks the starting point of the research, primarily involving historical stock prices over time and additional data. Following the data gathering process, I also conducted data combination for efficient storage. All data were acquired through APIs, making them easily reusable for future stock data acquisition endeavors.

All subsequent requests cover a time range from April 1, 2019, to April 30, 2023.

**3.1. Stock Prices Over Time**

Historical stock data is gathered by making requests to the IEX Cloud API, specifically using the financial data endpoints for 'historical prices.'

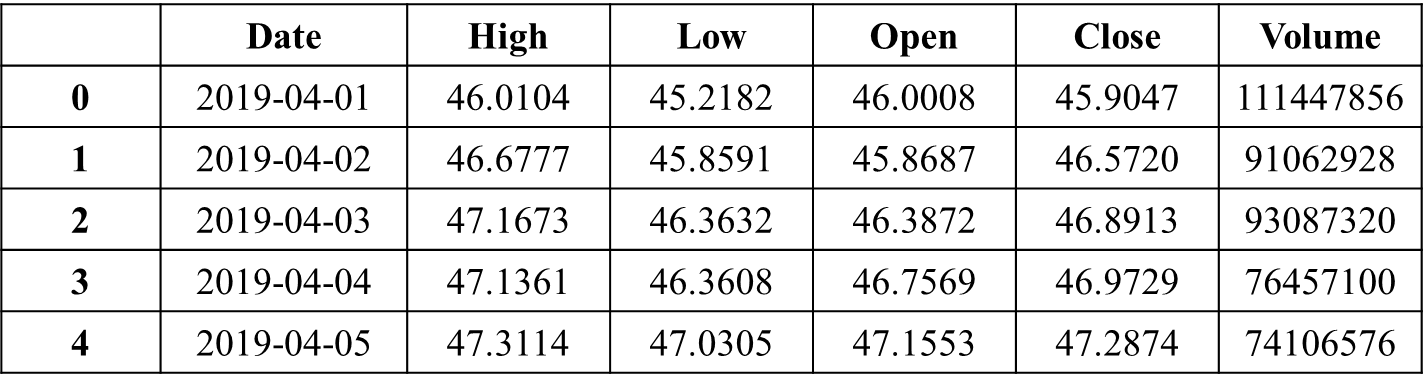


Table 1 Acquired AAPL Stock Data

The data acquisition function requires the ticker, API key, start date, and end date as parameters. By passing in 'AAPL' as the ticker and the API key to complete the URL, the function initiates a URL request that returns the data in JSON format. The necessary information is then stored using lists and loops, with date filtering implemented through pandas.

The final step involves returning the stock data in DataFrame format (Table 1) with columns named Date, High, Low, Open, Close, and Volume.

**3.2. Financial News Headlines**

The Efficient Market Hypothesis (EMH) states that stock market prices are largely driven by new information and follow a random walk pattern.[1] I collected financial news report headlines from both the New York Times and Financial Times. For the specific scenario of AAPL, I formed a query list using keywords 'aapl' and 'apple+inc' while implementing filter words containing 'aapl', 'iphone', 'ipad', 'mac', 'apple', and 'aal'.

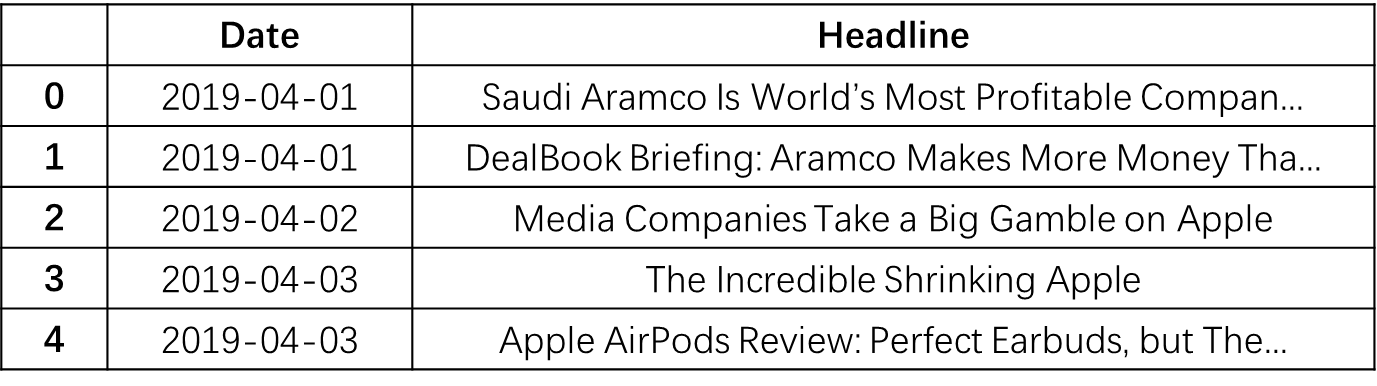


Table 2 Acquired Financial News Headlines

I initially obtained news data from the New York Times API, utilizing a main loop to iterate through date intervals and queries for API requests. To collect news headlines from the Financial Times, I used BeautifulSoup. The process begins by initializing headers and a Pandas DataFrame. Within the main loop, URLs are generated, HTTP requests are sent, and HTML is parsed. Subsequently, the obtained data was structured into a Pandas DataFrame, underwent cleaning, and was filtered based on predefined words.

Given that the data requests and gathering processes each take approximately 2 hours, I opted to store the acquired data in a CSV file for faster access.

**3.3 Sentiment Scores and Data Combination**

In the last step, I merged two news files into a single file and removed potential duplicates to facilitate sentiment data analysis.

The Sentiment Intensity Analyzer within the NLTK package was utilized to estimate sentiment scores, categorizing each dataset into four sections: compound, positive, negative, and neutral.[2] Then I aggregated news headlines for the same date by computing the average sentiment scores for each category and concatenating the headlines. Then I merged the sentiment scores and stock prices based on the date.

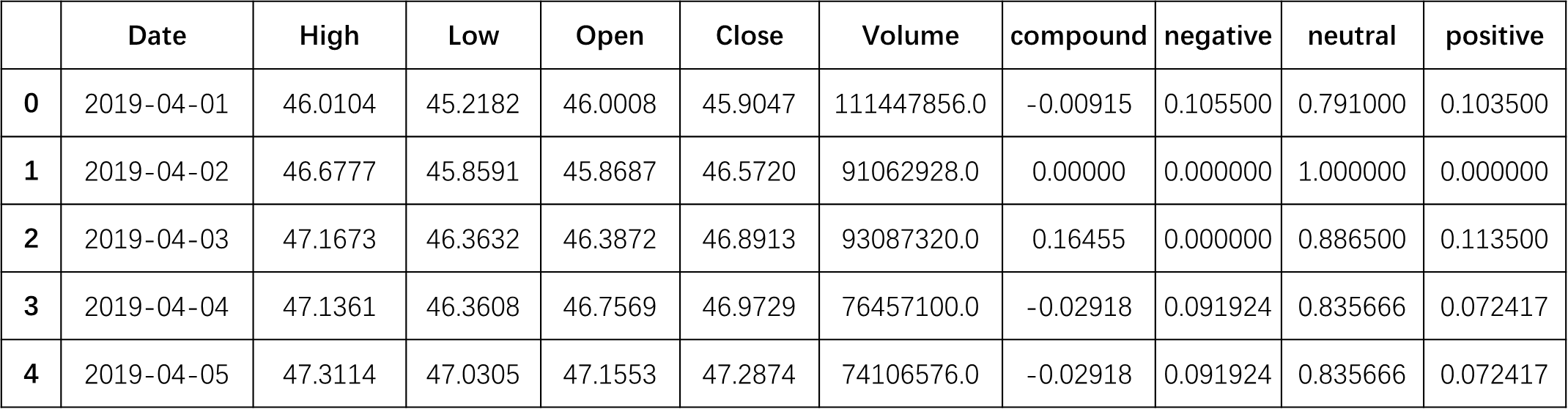


Table 3 Merged Date (Stock Price + Sentiment Scores)

To handle missing news data, NaN values for sentiment were set to the mean. Subsequently, the headline column was dropped, and any remaining Null values were removed.

The final dataset obtained is a merged version that includes date, stock prices over time, and news sentiment scores.

**4. DATA STORAGE**

Based on acquired data which have already been converted into DataFrame format, I used MongoDB Atlas for data storage in JSON format. And I implemented a local-based API server with FLASK for CRUD (Create, Read, Update, Delete) functions.

**4.1. Data Storage**

The connection to the database is initiated using the 'pymongo' library with a connection URI that incorporates credentials, cluster details, and additional parameters. This connection is verified by sending a ping and providing feedback on its status.

I created a database named 'AAPLdb' and a collection named 'StockData' within this database. Following this, I used the 'insert\_many' function to inject acquired data into the collection. This systematic process ensures the well-organized storage of the acquired data in JSON format in MongoDB, facilitating subsequent retrieval and analysis.

**4.2. API Implementation**

For further implementation, I created a Flask app served as a RESTful API for data management in the MongoDB database, providing CRUD operations. The main steps contain database connection, api endpoints identification and error handling.

The database connection is established using the connection string retrieved from the environment variable. Custom error handlers are implemented for HTTP 400, 404, and 500 errors, providing detailed error responses in JSON format. The concept of an endpoint is borrowed from HTTP network requests, with the corresponding relationships shown below.

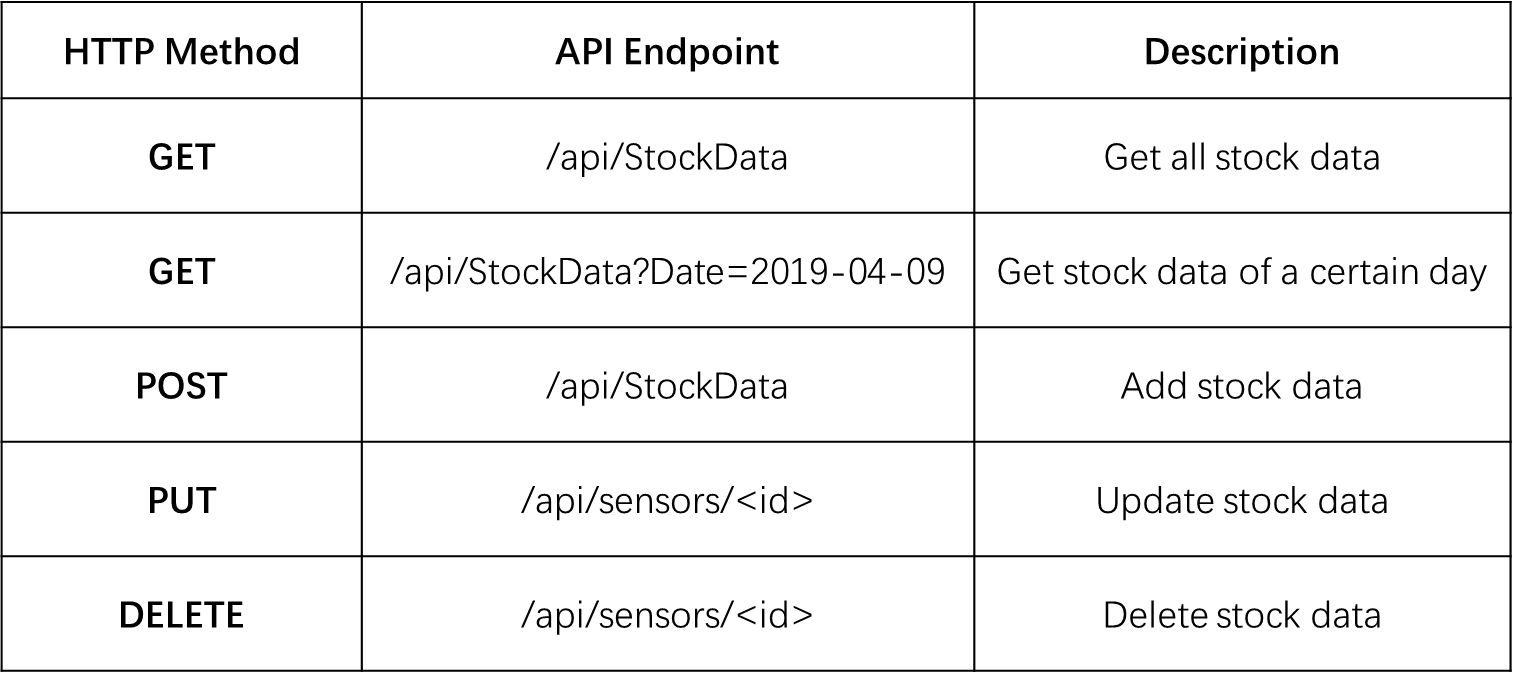


Table 4 API Endpoints Identification

To run the application, execute the script, and the server will start listening for requests on the specified port. All requests can be easily visualized using ‘Postman API Platform’.

**5. DATA Preprocessing**

After retrieving data from the database, essential preprocessing tasks include data cleaning, basic visualization for insights, and transforming data to align with analysis requirements. This ensures a reliable dataset, setting the stage for meaningful insights and informed decisions.

**5.1. Data Cleaning**

Data cleaning encompasses handling missing values and detecting/removing outliers.

1. *Missing Value Management*

In this phase, I checked data consistency, noting a discrepancy (1028 data points vs. 1489 date intervals) due to stock market closures on weekends and holidays. To ensure consistency, I used linear interpolation to fill missing values for dates in the DataFrame. This made the date sequence continuous, enhancing dataset coherence and reliability.

1. *Outlier Detection*

The Z-score serves as a valuable tool in identifying outliers within a dataset. It quantifies how far a data point deviates from the mean by standardizing its value. The z-score formula is expressed as follows:

A Z-score greater than 3 indicates an outlier, signifying a significant deviation from the mean compared to other data points.

This methodology helped me identify the wrong initial data I obtained where the stock data is in the raw format without adjustments for dividends. Specifically, 30 outliers were identified in a consecutive sequence. Upon obtaining the correct stock price data, no outliers were observed.

**5.2. Data Visualization**

Data visualization is a valuable means of gaining insights into a dataset. I created three visualizations representing fundamental data aspects: stock price, stock volume, and sentiment scores.

1. *Stock Variation Trend Over Time*

This figure illustrates the AAPL stock close price using a line plot and the day range using a shaded area plot, offering a visual representation of the variation trend spanning from April 1, 2019, to April 30, 2023.

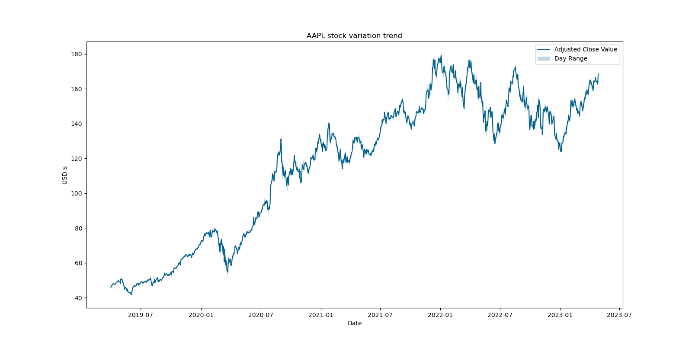


Fig. 1 AAPL Stock Variation Trend

1. *Stock Volume Changes Over Time*

This figure depicts the changes in stock volume on a daily basis over time.

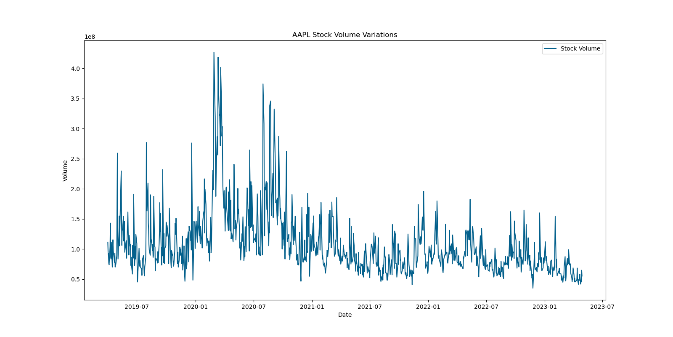
**

Fig. 2 Stock Volume Changes

1. *Sentiment Score*

This figure portrays the daily variation of sentiment scores, specifically focusing on the compound values, over the given time period.

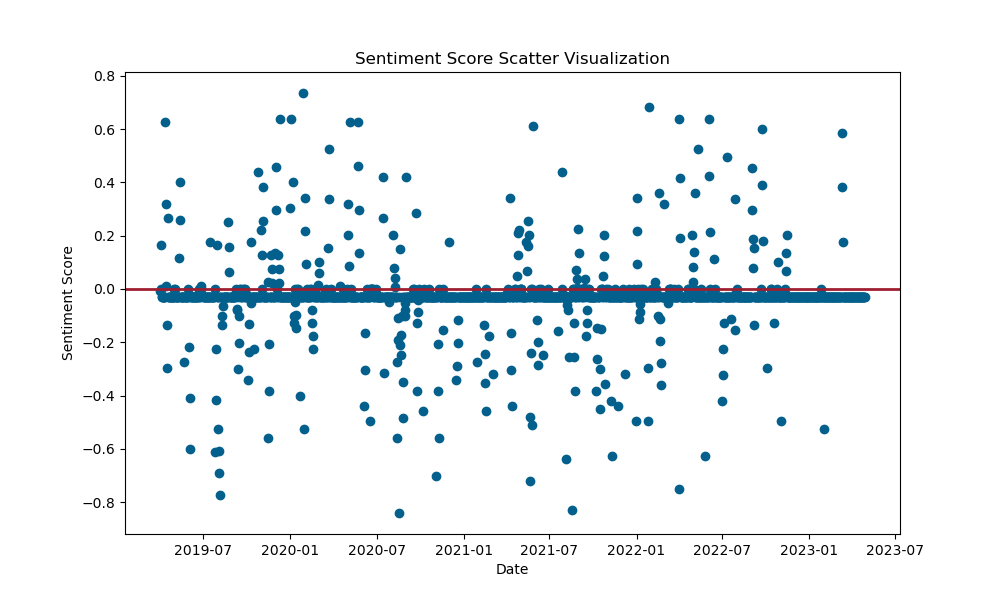
**

Fig. 3 Sentiment Score Changes

Based on the fundamental visualizations, it is observed that from 2020 to 2022, there is an upward trend in Apple Inc's stock price. Additionally, the daily trading volume in the year 2020 is notably higher compared to other periods. However, the sentiment score does not exhibit explicit signals, indicating the need for further exploration and analysis.

**5.3. Data Transformation**

Since the 'High, Low, Open, Close' values of the day exhibit a similar trend, I chose to retain only the 'Close' and 'Volume' columns in the stock data. The 'Compound' column in the sentiment data represents the aggregated sentiment scores, providing a meaningful representation among the polarity scores.

Therefore, the data selected for further transformation includes Date, the closing price of the day, trading volume, and the compound score.

**6. Data Exploration**

Utilizing well-preprocessed data and initial insights into the dataset, I applied various exploratory data analysis (EDA) techniques to gain a deeper understanding of the data. This section covers the analysis of the overall trend, unusual behavior, monthly return, seasonality, hypothesis testing, feature correlations, and financial indicator analysis. Each aspect is explored with detailed explanations, accompanied by plots and relevant formulas. The ARIMA model is employed as a powerful analysis tool in this section.

**6.1. EDA on Data**

1. *Overall trend with unusual behavior*

The visualization in section 5.2 revealed a slight downward trend in the beginning of 2020, followed by a dramatic upward trend for the remaining year. Additionally, there were notable fluctuations in trading volume during this period.

The figure (Fig.4) below illustrates the changes during this period, highlighting the impact of the COVID-19 pandemic on Apple stock.

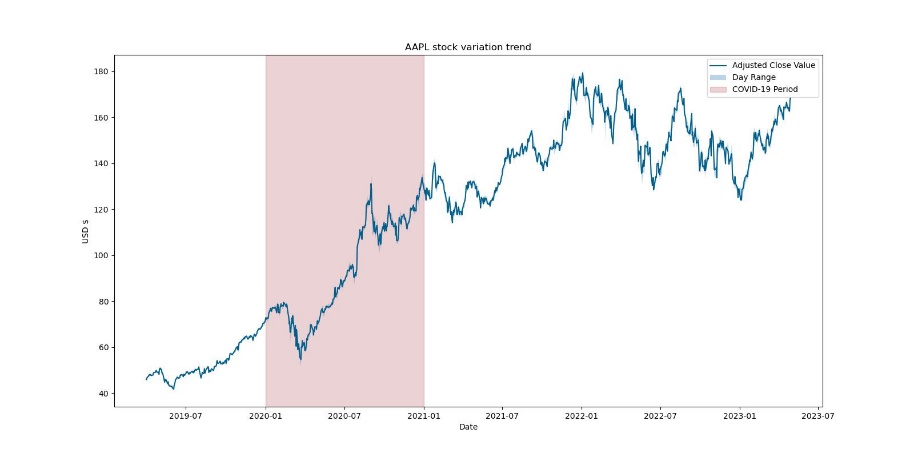


Fig. 4 Covid-19 highlighted stock trend

The lockdown measures and the surge in demand for electronic products, driven by the shift to remote work, significantly boosted the sales of Apple products. This highlights the influence of unusual incidents on stock trends.

1. *Monthly Return*

The monthly return process involves calculating the average percentage change in closing prices for a stock over distinct months. The monthly return for AAPL is presented as boxplot (Fig.5). The formula for the calculation is:

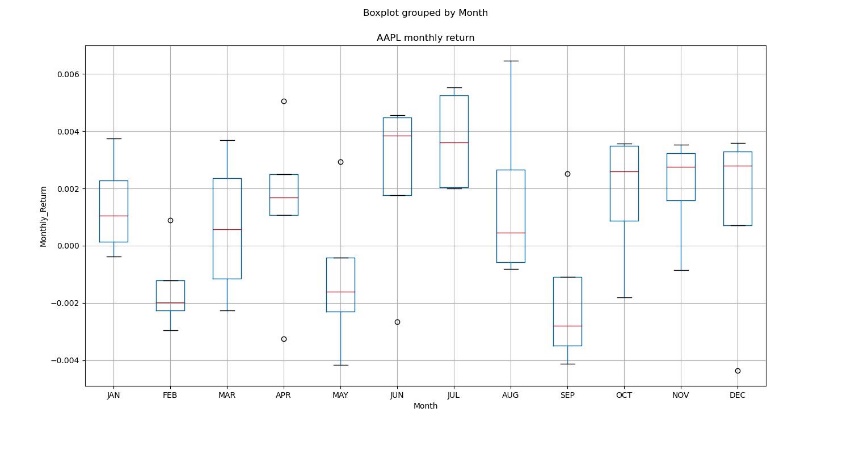


Fig. 5 AAPL Monthly Return

From the boxplot, I figured the following traits:

* The highest range of returns occurred in March and August.
* February and April exhibited a relatively small interquartile range (IQR), indicating less variability in returns during these months.
* April, June, July, and December showed a low downside trade probability, with the minimum line close to the 25th percentile. This suggests that these months might provide favorable opportunities for entering the market with lower downside risk.

1. *Hypothesis Testing*

I employed the Augmented Dickey-Fuller (ADF) test for hypothesis testing. The null hypothesis assumes the presence of a unit root, signifying that the data is non-stationary. The ADF test calculates the ADF statistic and its associated p-value.[3] Critical values are provided for comparison; if the ADF statistic is more negative than these critical values, the null hypothesis is rejected.

In this case, the calculated ADF Statistic is -1.238053, the p-value is 0.657016, and the Critical Values are 1%: -3.435, 5%: -2.864, 10%: -2.568.

Given that the p-value exceeds 0.05, and the ADF Statistic is not more negative than the critical values, the null hypothesis is not rejected. Consequently, the data is deemed non-stationary, indicating that the stock close price exhibits a certain degree of trend or seasonality.

1. *Seasonal trend analysis*
2. Seasonal Decomposition

Through the analysis of the null hypothesis, I aim to uncover seasonal trends using time series decomposition using additive model. The components are outlined below:

where is the data, is the seasonal component, is the trend-cycle component, and is the remainder component.[4] The figure shown below represents the result of seasonal decomposition.

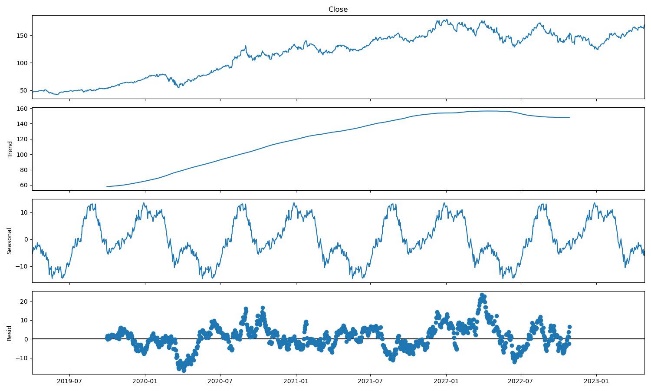


Fig. 6 Seasonal Decomposition Components

Based on the figure presented above (Fig.6), the general trend displays an overall increase over time, with a minor deviation in the middle of the year 2022. Concurrently, the seasonal pattern exhibits a repetitive cycle, alternating between valleys and peaks.

1. Seasonality of a specific year

To delve deeper into the seasonal trend of the year, I'll use the example of the year 2020. In time series analysis, the moving average (MA) serves to smooth short-term fluctuations while highlighting long-term trends. Here, I've chosen a 20-day moving average for the analysis.

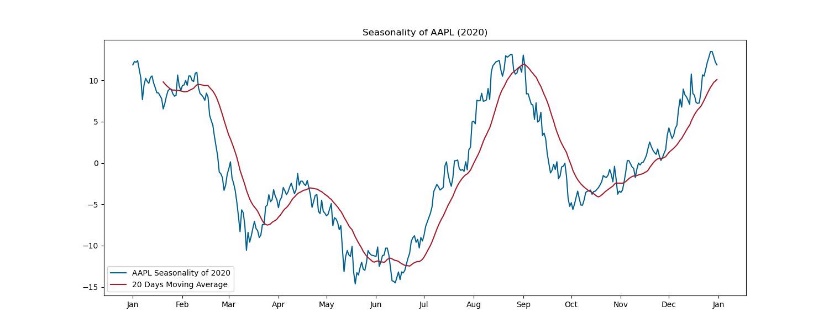


Fig. 7 AAPL’s seasonality of 2020

The chart displayed above (Fig.7) illustrates the seasonality trend resembling a wave, with significant highs occurring around September and notable lows around June.

A straightforward strategy could involve buying in June or July and selling in September to align with these recurring patterns.

1. Seasonality Adjustments

Given that an additive model was utilized to identify seasonality, adjustments based on seasonality can be made to the stock data. For comparison, I selected the year 2020, which exhibited an upward trend (Fig.8), and the year 2022, showcasing a relatively ordinary market (Fig.9).

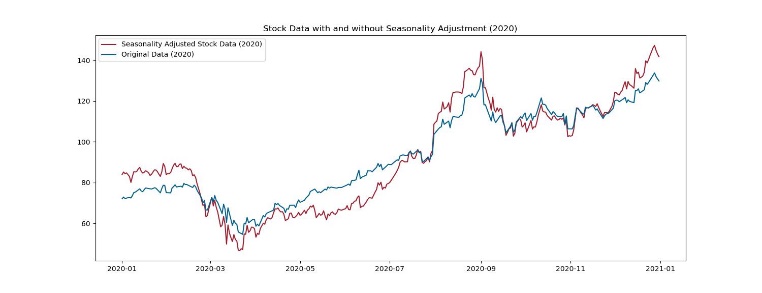


Fig. 8 Seasonal Adjustment (2020)

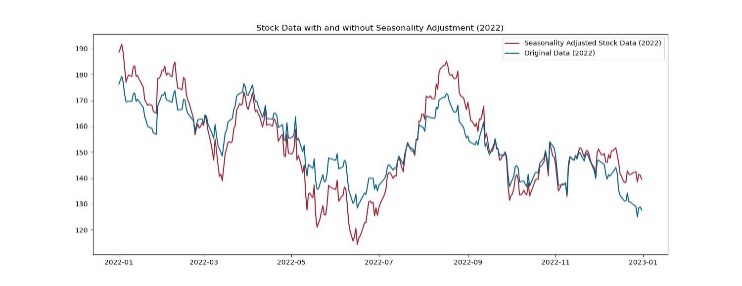


Fig. 9 Seasonal Adjustment (2022)

In periods of strong bull or bear (upward or downward) trends, the effects of seasonality may be less observable.[5] However, in a range-bound market, these effects can become more evident. The seasonality effect is more pronounced in 2022 compared to 2020, indicating its significance during periods of market stability.

1. *Feature correlations*

In recent research, the correlation between the stock market and news sentiment is expected to be evident. However, capturing this relationship is challenging due to the limited amount of related news that was gathered (Fig. 3). The heatmap below illustrates the correlation between sentiment scores and stock close prices.

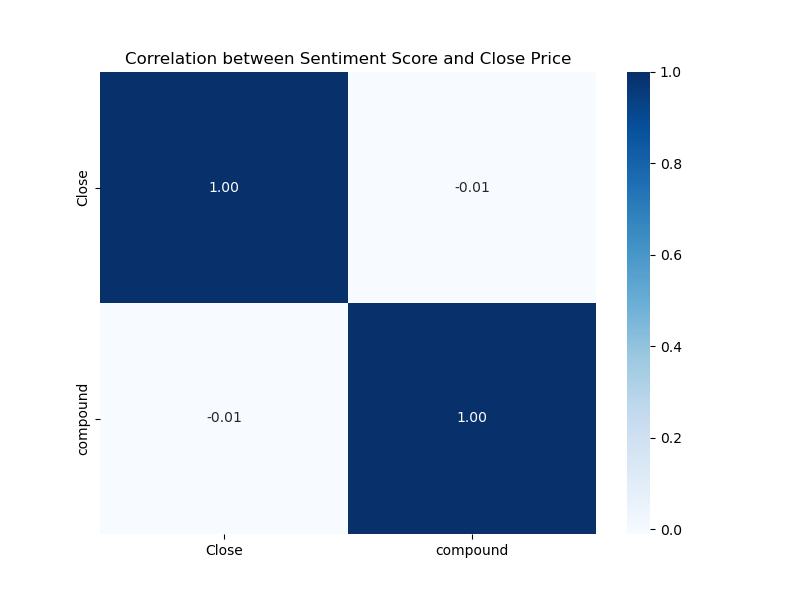


Fig. 10 Correlation between Sentiment Score and Close Price

**6.2. EDA on Known Financial Indicators**

The On-Balance Volume (OBV) is a technical indicator that gauges buying and selling pressure by cumulating volume changes corresponding to price movements. Positive volume is added when prices rise, and negative volume is subtracted when they fall. Exponential Moving Average (EMA) is often used for trend smoothing. Crossovers between OBV and its EMA signal potential buying or selling points, aiding traders in decision-making based on volume and price dynamics.[6]

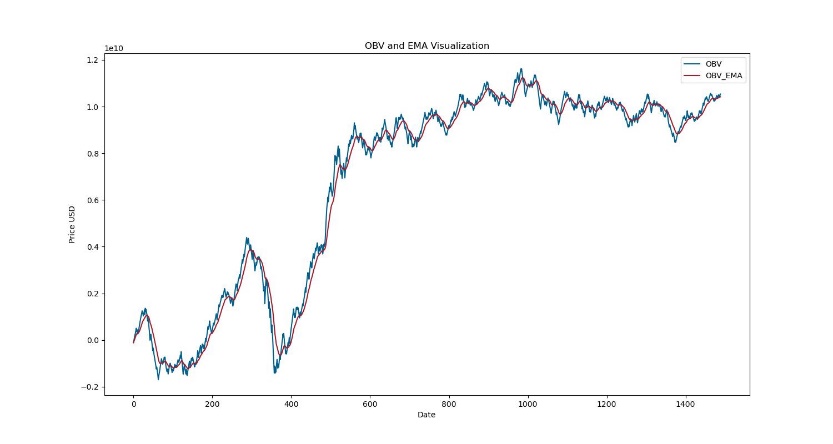


Fig. 11 OBV and EMA of AAPL

When the OBV is above its EMA, it generates a buy signal, and conversely, a sell signal is triggered when OBV is below EMA. I then visualized these signals across different time periods in the dataset.

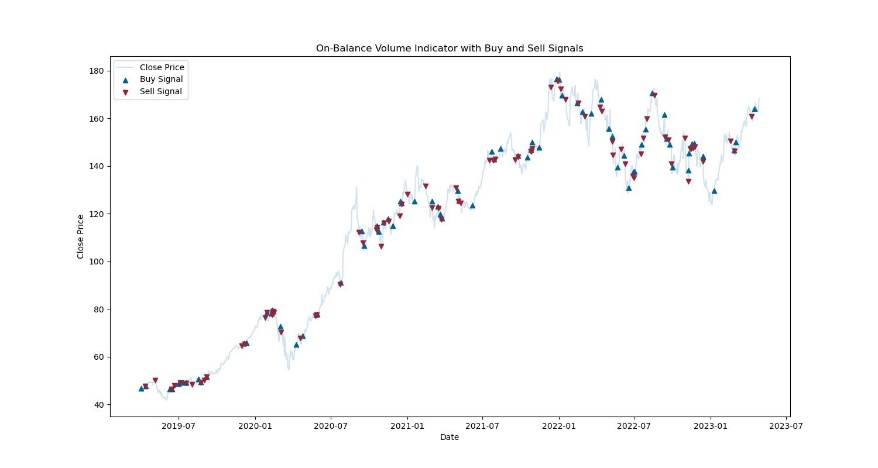


Fig. 12 OBV Based Buy and Sell signals

**7. Forecasting**

In this section, Price Forecasting is conducted using Long-Short Time Series (LSTM) with the objective of predicting prices from April 1, 2023, to April 30, 2023, based on historical data.

I employed LSTM for forecasting, integrating indicators generated in the previous section. The model was constructed with 3 LSTM layers and included Dropout regularization for optimal performance.

In this analysis, I compared the performance of two training models (Model1 and Model2) based on different time ranges, starting from April 1, 2019, and January 1, 2021, respectively. Additionally, I assessed the impact of incorporating sentiment scores in the forecasting process (Model 3) within the same time range. The forecasting utilized a window size of 20 days, with Root Mean Squared Error (RMSE) serving as the metric for training losses. The results illustrated a close alignment between the predicted curve and the original data, showcasing the effectiveness of the forecasting models.

The results of the models are shown below:

|  |  |  |
| --- | --- | --- |
|  | Training Start Date | RMSE |
| M1 | 2019.4.1 | 4.2692 |
| M2 | 2021.1.1 | 2.9015 |
| M3 | 2021.1.1 | 2.5113 |

Table 5 Forecasting Results

**7.1. Development of Model Using Stocks**

LSTM serves as a robust deep learning technique for time series forecasting, leveraging date and close price data in this context.[7] The dataset was systematically divided into training and testing subsets, integrating a specific lookback window of 20 days.

The constructed LSTM model consists of three layers with tanh activation and incorporates Dropout layers. Additionally, it is compiled using the Adam optimizer. The training process involves fitting the model to the training set for 200 epochs, utilizing a batch size of 64, and allocating 5% of the training data for validation. This comprehensive approach enhances the model's ability to capture and forecast underlying patterns in the time series data.

Initially, I applied the model to the entire dataset spanning from April 1, 2019. As the overall trend exhibited variations, I refined the data by focusing on the period after 2021, where a smoother trend was apparent. This adjustment resulted in a notable reduction in Root Mean Squared Error (RMSE), decreasing from 4.2692 to 2.9015(Fig.13).

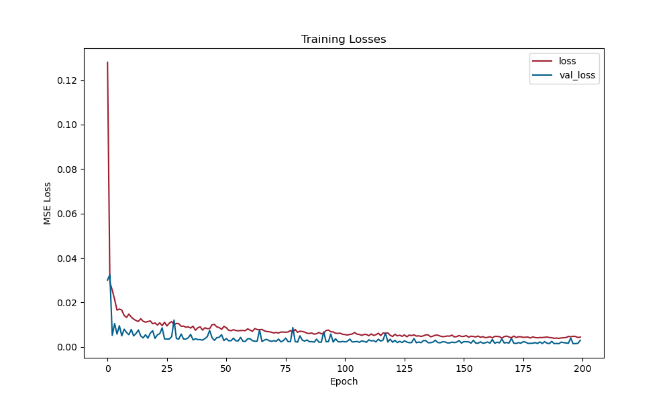


Fig. 13 MSE Loss During Training Process

The prediction result of Model1(Fig.14) and Model 2(Fig.15) is shown below, which both exhibits a well-matched curve. And the optimization can be observed through the comparison with original data.

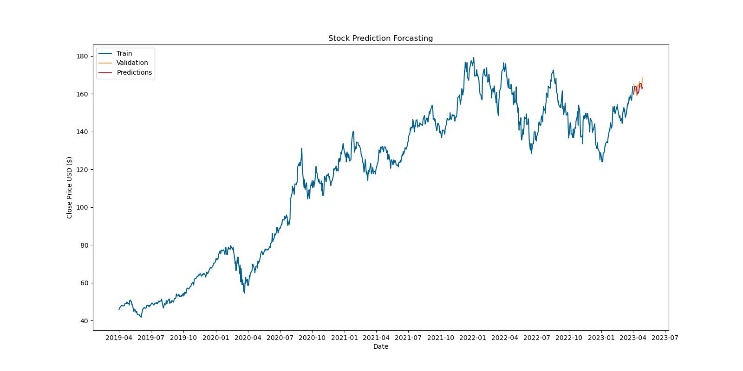


Fig. 14 Prediction Curve of Model 1

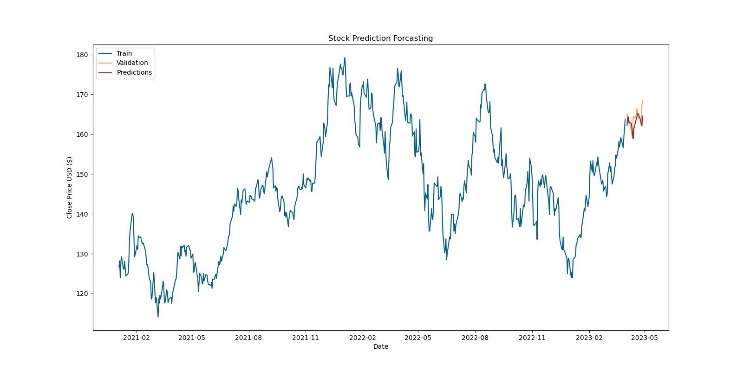


Fig. 15 Prediction Curve of Model 2

**7.2. Development of Model Using Stocks and Sentiment Scores**

In this section, the analyzed sentiment compound score is integrated into the prediction process alongside close price, utilizing the same LSTM layers.

No observable relations between close price and sentiment score were found in the previous correlation analysis. However, with the assistance of sentiment scores, the RMSE has been reduced from 2.9015 to 2.5113, revealing additional significance in the context of sentiment analysis.

The prediction curve of Model 3 is shown below:

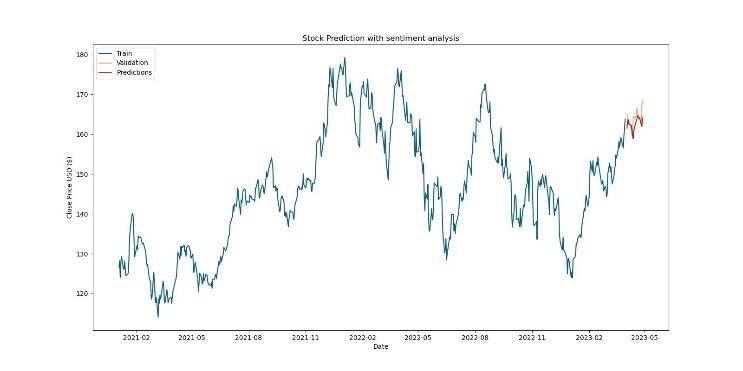


Fig. 16 Prediction Curve of Model 3

**7.3. Implementation of Evaluation Metrics**

Overall, the model incorporating sentiment scores and utilizing training data from January 1, 2021, exhibits the best performance. Evaluation is measured using the Root Mean Squared Error (RMSE), calculated as the root of the mean of squared errors between predicted and actual values.

To further assess certainty, I employed ARIMA for exploration, visualizing the results with a 95% confidence interval (Fig. 17). The confidence interval becomes more ambiguous with the length of the training set, indicating that the models have limited capability in predicting closing stock prices significantly far into the future.

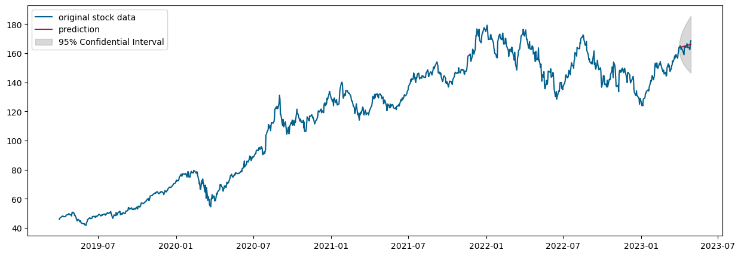


Fig. 17 Stock Prediction Line with 95% Confidential Interval

Further improvement can be achieved by enhancing the sentiment analysis model with additional data points from diverse sources. Additionally, applying seasonal trend adjustments could contribute to generating more robust and concrete results in the forecasting process.

**8. DEcision making**

Decision making is the final objective for the entire process which is trying to maximize the overall profits. The principle is buying at lower prices, selling at higher prices, and holding stocks during periods of smooth fluctuations.

Further analysis of the prediction results for April 2023 reveals detailed insights, as illustrated in Fig.18.

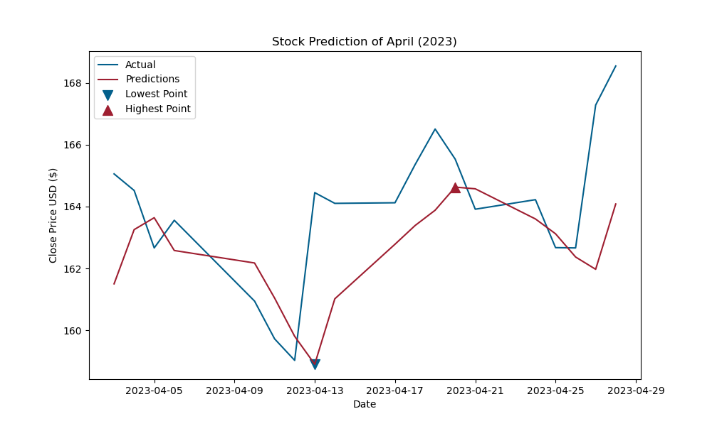


Fig. 18 Detailed Close Price Prediction of April, 2023

I established a trading algorithm to deicide buy and sell signals with future prediction and return threshold.

*Algorithm Explanation:*

*if buy return>=sell return and buy return>=return threshold: Buy*

*elif sell return>buy return and sell return>=return threshold: Sell*

*else: Keep*

Given the limited prediction time range, I've set the expected return threshold at 25%. The signals are derived from predicted values and overlaid on the actual data curve. Notably, the sell signals predominantly occur at relatively high levels, underscoring the effectiveness of the decision-making algorithm. However, due to the close price being consistently high throughout the month, entering the market may not be advisable so no buy signal is generated, as depicted in the figure below (Fig. 19).

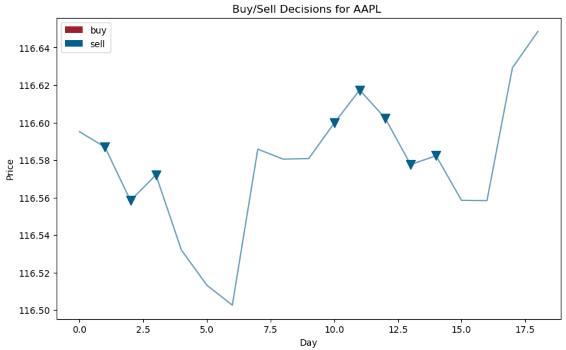


Fig. 19 Buy and Sell signals in April, 2023

Look further into the OBV indicator (Fig.20), fewer signals are generated in this period.

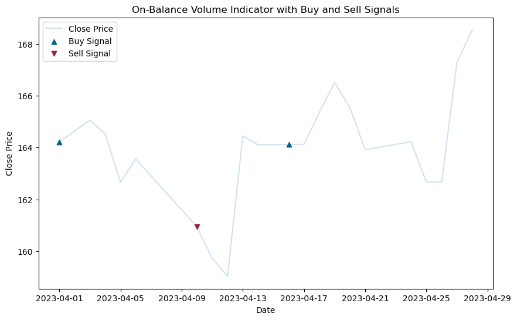


Fig. 20 OBV indicator in April, 2023

With a $10,000 budget, I recommend allocating $3,000 to buying on the 5th, $5,000 on the 12th, selling $5,000 on the 19th, and selling another $5,000 on the 28th .

Predictions are pivotal in the decision-making process of buying, selling, or holding stocks. They are influenced by forecasted future prices, sentiment trends, and computed financial indicators. In longer-term predictions, accounting for seasonality becomes a valuable aspect in trading. These factors collectively contribute to a comprehensive decision-making process in the realm of stock trading.

**9. Conclusion**

In summary, the project successfully combines deep learning techniques, sentiment analysis, and financial indicators to analyze and forecast AAPL stock. While seasonality and monthly return provide a macro view, they are not obvious in strong bull or bear market trends. For accurate trend capture, prediction based on historical data plays a predominate role with the assistance of current public sentiment. Decision-making is guided by meaningful indicators such as return threshold and on-balance volume. Indeed, real-life stock fluctuations are influenced by numerous factors, making it challenging to accurately predict future directions. As a result, stock analysis and predictions hold guiding significance primarily in the short term.

Challenges include data limitations and predicting within a constrained time range. The study showcases the potential of LSTM models and sentiment analysis in enhancing stock price predictions. Future improvements could involve acquiring more diverse sentiment data and refining decision-making algorithms for extended prediction periods. Overall, the project offers valuable insights into the integration of advanced analytics for informed stock market decisions.

**10. References**

1. Kalyani J, Bharathi P, Jyothi P. Stock trend prediction using news sentiment analysis[J]. arXiv preprint arXiv:1607.01958, 2016.
2. Elbagir S, Yang J. Twitter sentiment analysis using natural language toolkit and VADER sentiment[C]//Proceedings of the international multiconference of engineers and computer scientists. 2019, 122: 16.
3. Christopoulos D K, Tsionas E G. Financial development and economic growth: evidence from panel unit root and cointegration tests[J]. Journal of development Economics, 2004, 73(1): 55-74.
4. Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on <2024.1>
5. Pagan A R, Sossounov K A. A simple framework for analysing bull and bear markets[J]. Journal of applied econometrics, 2003, 18(1): 23-46.
6. Granville J E. Granville’s New Key to Stock Market Profits[M]. Pickle Partners Publishing, 2018.
7. Shen J, Shafiq M O. Short-term stock market price trend prediction using a comprehensive deep learning system[J]. Journal of big Data, 2020, 7(1): 1-33.

**APPENDIX: MINI CASE STUDY**

**Title: Enhancing Fairness in TikTok's Recommendation System based on Age**

***Student ID: 64***

The central aim of nearly every recommendation algorithm on social media platforms is to prioritize content based on the likelihood of user engagement. For instance, YouTube places emphasis on click-through rates, Netflix focuses on user ratings for movie series, while TikTok's algorithm uniquely considers a combination of user actions like liking, commenting, play time, with particular emphasis on video completion rates as a robust indicator of user engagement.[1]

Platforms have begun integrating fairness considerations for creators to address user biases and prevent algorithms from amplifying them without intervention.[2] As a data analyst at TikTok, I will conduct an analysis on age fairness to improve the recommendation system, ensuring fairness and inclusivity for all users.

***Relevance to use case:***

Age serves as a valuable indicator of distinct habits, hobbies, preferences and daily focus. Understanding age-related behaviors is crucial for tailoring content that resonates with users, addressing their specific interests and priorities and promote fairness within specific age groups.

The mainstream population on TikTok falls between 18-34 years old, with those under 25 more engaged in current trends and pop culture content, while those over 25 may concentrate more on working professions. The gradual ascent of older users is more engaged in educational, lifestyle content.[3] By understanding and tailoring content recommendations to these age-specific preferences, TikTok can ensure that users from both groups receive content that resonates with their daily lives.

***Metrics and Fairness Criterion:***

***Metrics:***

***Content Relevance by Age:***

This metric assesses the relevance of recommended content based on the user's age by analyzing whether recommendations align with the preferences or interests of users in different age groups using metrics such as views, likes and shares for the recommended content. It involves analyzing whether recommendations align with the preferences or interests of users in different age groups caused by hobbies, daily habits and etc.

Significant variations in content popularity may suggest that the algorithm favors certain age groups, impacting the diversity and inclusivity of the content shown to users.

***Engagement Time for recommended content:***

For short-video platforms like TikTok, user engagement time serves as a crucial metric for evaluating potential variations in how various age groups interact with content suggested to them on the 'For You' page.

Substantial differences in engagement time across age groups may signal that certain recommendations are more appealing to one age group, indicating potential bias.

***Fairness Criterion:***

***Balanced Exposure and Engagement:***

This criterion aims to ensure that contents receive a similar level of exposure and engagement where users with diverse characteristics and preferences coexist. It avoids biases in the recommendation system that might result in overemphasizing certain creators or content types. This approach aims to mitigate biases, fostering a recommendation system that caters equitably to users with varying preferences and characteristics, promoting inclusivity and diversity on the platform.

***Integrating into the machine learning pipeline:***

To integrate the improvement into recommendation pipeline, adding regularly assess to the model's fairness using the defined metrics (Content Relevance by Age, Engagement Time for Recommended Content) to identify and rectify any biases.

To implement algorithmic adjustments based on the fairness criterion, this requires recalibrating recommendation scores to ensure content exposure and engagement are more evenly distributed across different age groups.

In summary, prioritizing fairness in TikTok's recommendation system based on age is crucial for an inclusive user experience. Metrics like Content Relevance by Age and Engagement Time for Recommended Content illuminate potential biases and guide adjustments, while the fairness criterion of Balanced Exposure and Engagement ensures diversity in content representation. Integrating these measures into the machine learning pipeline through regular assessments and algorithmic adjustments establishes a continuous improvement cycle, fostering an unbiased and engaging content experience for users of all ages.

1. Narayanan, Arvind. "Understanding Social Media Recommendation Algorithms." (2023).
2. Geyik, Sahin Cem, Stuart Ambler, and Krishnaram Kenthapadi. "Fairness-aware ranking in search & recommendation systems with application to linkedin talent search." Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining. 2019.
3. MegaDigital. (n.d.). TikTok Age Demographics. MegaDigital. Retrieved Month Day, Year, from https://megadigital.ai/en/blog/tiktok-age-demographics/