Apple Stock Price Prediction using NLP and Time Series Models

Yao Zhang - yz4481
Catherine Liu - xl3239
Shuobofang Yang - sy3113
Xianglin Lu - xl3258
Justin Ingardia - ji2278

Introduction

The conventional belief in the stock market is to “buy the rumor, sell the news”. You are bound to make a fortune if you were right on the speculation. In our preliminary research on conventional belief, we hypothesize that there is a strong tie and reaction to the company-related news on the day of its release. We examined the fact by collecting all news articles from Benzingas related to Apple, then breaking them down into a dictionary and weighing each word. We used the percentage of the change (open and close price) as our output. Then we compared whether there is a relationship between the keywords and the price change percentage. We have also introduced a time series model with various methods to benchmark the performance of models and found the best models to predict the changes and the price of Apple stock.

Objectives

1. Use sentiment analysis on news text data using natural language processing to predict future stock prices after 30 trading days.
2. Find the best model for time series in the yearly forecast

Research questions

- What is the impact of company-related news on Apple's stock price on the same day of its release?
- Which keywords in the news articles related to Apple have the strongest association with the percentage change in its stock price?
- How can we identify the most suitable time series model for Apple stock, considering the stock’s characteristics and performance over time?

Steps of fetching and preprocessing data

The time of the dataset: 2010-01-04 to 2023-03-23

Prepare Stock Data

1. Downloaded AAPL's historical stock price and volume data directly from Yahoo Finance.
2. Data cleaning for prediction
3. **For NLP:** Combined with natural language variables to perform analysis
4. **For time series:** Converted to time series, xts, and zoo object.
5. Benchmarked performance.

Prepare News Data

1. We used Benzenga's restful API to download the historical News data. As background knowledge, Benzinga is a timely financial news and analysis service that publishes reputable news and analytical articles. Benzinga's News is actively and timely posted on commonly used brokers which may affect investors' decisions.
2. A Free API key is obtained after registration on Benzinga's developer website. With an API key, we called and got the response JSON data and then converted it to data frame and CSV. There is a limit of 100 news articles fetched per API call, so we also managed to use a for loop to combine all the responses. This part is written in a Python notebook, where a Benzinga package is needed to deal with Benzinga's API efficiently.
3. Selected wanted and useful columns from the raw data.
4. After observing the news data, the News title without Apple or its alias has little relevance to Apple company's corporate performance and market condition, so we decided to remove those News.
5. Cleaned the News' Body column by parsing the text and removing the HTML tags.

Preprocess Data for Training

In preparation for our data, we cleaned the data from HTML text and parsed it into raw text format. We have extracted all the news between 2010-01-04 to 2023-03-23 on the Benzingas network that is related to Apple Inc(AAPL). We downloaded data from Yahoo Finance on the stock price to match the days of the stock market. We only looked at weekdays data and ignored holidays. Since we merged the two CSV files based on the date, therefore, on the final data table are all active trading days.

How did we deal with the missing data?

After we joined the trading table from Yahoo and the news table from Benzinga by the date, it inevitably resulted in many null/NA values. It is because Benzinga has News published on both weekdays and weekends, by contrast, the market does not open on the weekends or during the holidays.

The way we input these null values is to impute from the nearest News data or trading data using the fill function from tidyr library. To be more specific, if the news is released when the market is closed, the news were put to the last trading day, and we assumed it takes effect on the next trading day achieved by this code:

```
news_price[cols_to_fill] <- news_price[cols_to_fill] %>%
        tidyr::fill(everything(), .direction = 'down'), .direction = 'down')
```

And for the case when the market is open, but there was no news on that day, the null value is filled with the last available observation achieved by this line:

```
news_price <- news_price %>% tidyr::fill(everything(), .direction = 'up')
```

Then any duplicated lines were removed.

Natural Language Processing

In our NLP model, we developed four approaches that are commonly used in encoding text data into numerical features.

1. Firstly, we calculated the term frequency and inverse document frequency of both the body text and the title text and sum it up for each trading day. The steps include calculating the TF score of each word throughout the corpus, the IDF score of each word throughout the corpus, the production of TF and IDF for each word, and the sum of the production in each article.
2. Secondly, we embedded the words in each article into vectors (numerical features). And it follows these steps: Created a vocabulary of words. Trained word embeddings using GloVe (unsupervised learning algorithm), one of the most popular word-embedding models. Calculated mean word embeddings for each article. Calculated the max, sum, and mean of the embeddings.
3. Thirdly, we embedded the characters into vectors. It follows the following steps: Converted text data to sequences. Padded sequences to a fixed length. Created an embedding layer that maps the integer-encoded characters to dense vectors.
4. Lastly, we explored using multiple lexicons that catered toward financial analysis to benchmark results. We tokenized the words, and output sentiment scores of all three lexica, “afinn”, “bing”, and “nrc”.

Since we didn’t want to predict the stock price by a particular article, instead, we grouped the populated vectors by date and integrated the vectors by their mean. The reason we took the average of the articles on each day is that we wanted to avoid the

underlying bias of a particular article. Taking the average of all the articles enables us to consider multiple aspects of the news, and thus generates a wholistic numerical representation.

After stemming and tokenizing our text data, we were able to answer the question:

- Which word is the most correlated with ups, and which is most correlated with down? Which keyword has the strongest association?

In our analysis, we found “x” keyword has the strongest association with the rise in stock price. On average, when the word appears “however many time”, the stock generally rise x percentage. “Xxxx” is correlated to a sell in the market, when “xxx” appear frequently, the stock prices often decrease.

After outputting the “x” variables, we implement a assemble method of “Catbboost” which is known to be a great tool for text analysis since often it might be high dimensional text data. By doing this assemble method, the results of the model were able to produce an RMSE of xx in the test set.

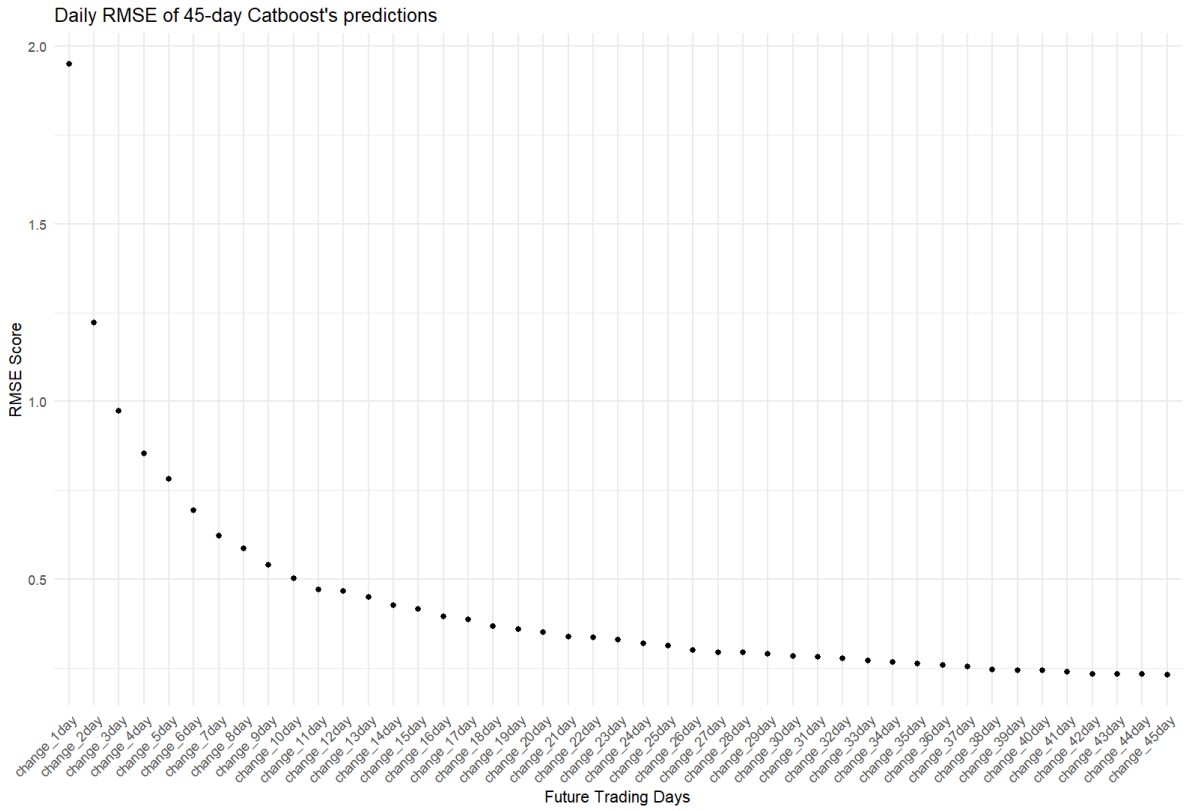
Regression Models

Following the natural language processing where we populated numerical features as the vectors that replace the text data, we were then able to use these features as independent variables, and stock price’s future change as dependent variables to build the supervised regression models.

As mentioned in the previous section, the dependent variable is generalized by the stock’s future change from 1 day to 45 days using the lag function on historical price data. We did this because we didn’t know the length of time that sentiment of the news will take affect the stock price, so we managed to compare the prediction of 45 models that trained from these dependent variables to compare the error rates.

Using Linear Regression and CatBoost models, we trained, tuned, and validated the supervised learning model using text features as predictors and price movement as the dependent variable. And as expected CatBoost outperforms Linear Regression models by comparing with the loss function.

We did not train a greater amount of the supervised learning models such as Random Forest, XGBoost, RNN, and LSTM because lowering the result of the loss function and maximizing the accuracy of the model is not the priority of our project. Instead, we would like to examine if the News features work well as predictors to some extent.



The previous plot compares the metrics of the CatBoost model's prediction on future price movements from day 1 to day 45. The mechanism we used to decide which to use as our model is something like Elbow's rule, we kept the change in 30 days model because of its low error rate and yet did not excessively fit the training data. However, the other models trained from 1 day to 45 days also have the potential to perform better than the model we chose.

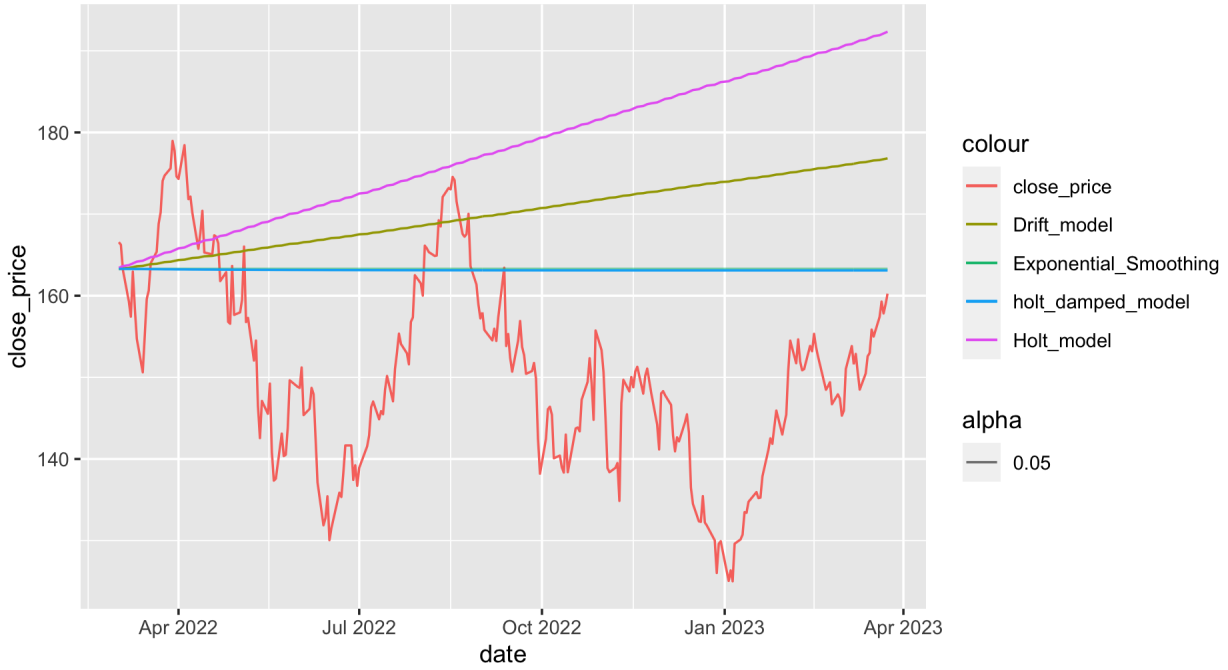
Time Series Models

In our time series analysis of Apple (AAPL) stocks, we used a range of models including the prophet, drift, exponential smoothing, holt, holt-damped, auto-arima, and the average model. Our objective was to predict the close price of the stock for 365 days, using data split into a training set comprising data up to 03-01-2022, and a test set comprising data from 03-02-2022 to 03-23-2023. Our training set consisted of 3061 trading days, while the test set comprised 268 trading days.

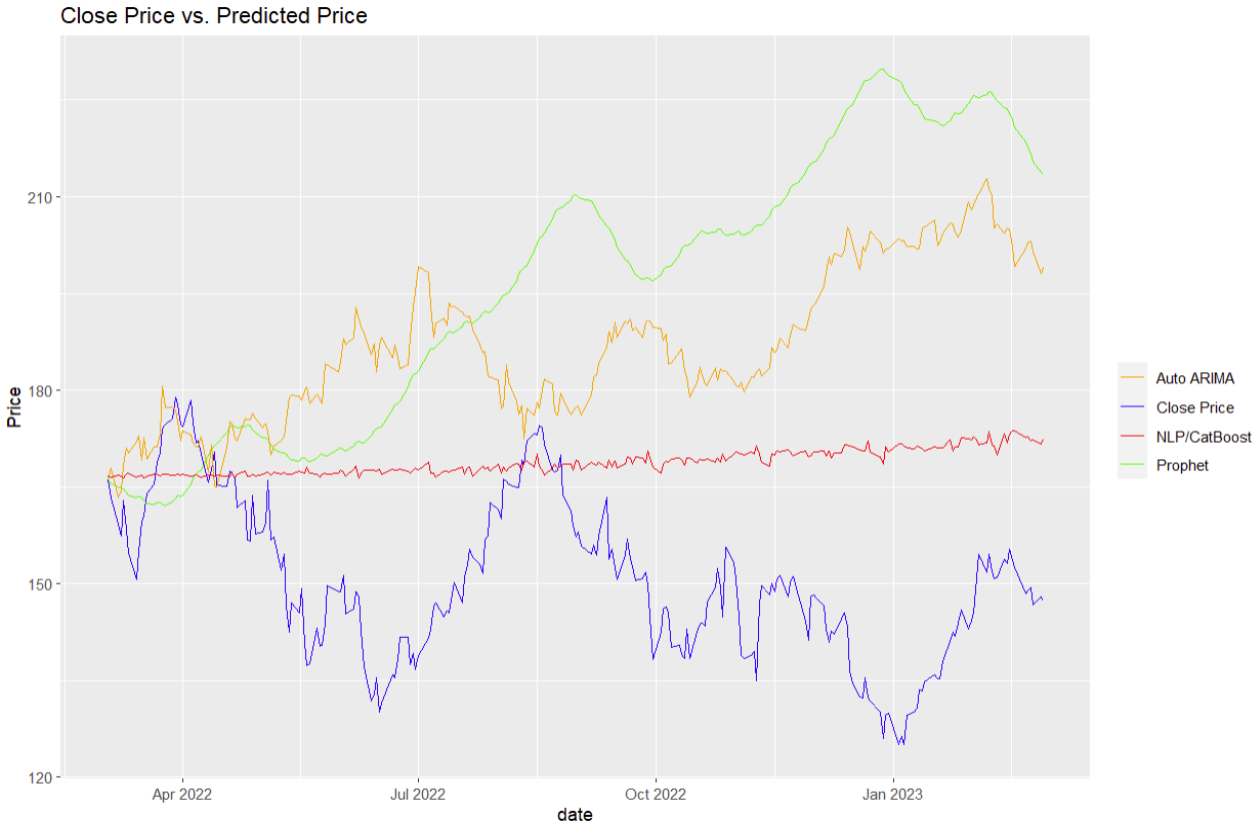
We chose to use the average model as a benchmark, despite its limitations in predicting stock prices accurately. The reason for this was to compare the performance of more sophisticated models against a simple baseline. We then applied various time series models within the “forecast” library, with each model trained on the training set and tested on the test set.

Of the models tested, our best-performing models were the Holt-damped model and exponential smoothing, with RMSE of 17.01 and 17.14, respectively. These models were able to accurately capture the trend and movement of Apple stock prices, and their performance was superior to the other models tested.

holt_damped_model	ses_model	drift_model	auto_arima
17.00930	17.14425	23.73158	23.81542
holt_model	prophet_model	average_model	
32.03373	53.01230	107.09136	



Despite the prophet model not performing as well as the other models, it was still able to capture some of the up-and-down trends of the stock price when visualized. This model is known for its ability to adapt to trends, and it is also capable of adapting to uncertain intervals and significant events. With further tuning and adjustment, it could potentially become a strong model for mimicking the movement of the Apple stock.



Recommendation/Conclusion

It is important to note that the stock market is highly complex and subject to numerous external factors that can influence stock prices. While our models provide valuable insights into stock price prediction, their performance may vary depending on the specific context and the variables considered. Therefore, it is important to continually update and refine the models based on the latest market trends and data and to carefully select the most relevant features and variables for accurate predictions.

However, we do believe our models were only able to provide some perspective in valuing the price, especially in the short term. From our findings, we would recommend other combinations of methods be tested as well. Starting from data gathering, feature selection, and even using a different time could all have different results and react differently with different models. Without the computation power and manpower, our project can experiment with limited models and a batch of data. Within the models themselves, they can also be tuned for a better result that we could not explore with our

dataset. Therefore, tuning in the models and adjusting the variables to optimize results is what we or others could do benchmarked using our models.

Our time series analysis of Apple (AAPL) stocks using various models highlights the importance of experimenting with sophisticated models that are capable of accurately capturing the trend and movement of stock prices. While some models may not perform as well as others, they can still provide valuable insights into the up-and-down trends of the stock price.

Through our research, we found that our model might perform better with stock prices that are captured in intervals of seconds, or even smaller time units. In the future, we could also experiment more with other supervised learning models to train our data and make predictions. We would also recommend when exploring the stock market, other news sources could be gathered, and more data and a high dimension model that could be employed to understand complex data and features. However, it does depend on case to case basis.

Use Cases

Mid-term and long-term trend prediction: By utilizing various types of Time Series models in our project, we were able to forecast stock price trends over medium to long-term periods. These predictions can be valuable for investors looking to capitalize on market trends and make informed investment decisions. The models helped us identify potential profitable investment opportunities by analyzing mid-term and long-term trends. By understanding these trends, investors can make more informed decisions on when to enter or exit the market, thereby maximizing their returns on investments. Additionally, these models can be used by portfolio managers and financial analysts to assess the performance of individual stocks and the overall market, enabling them to make strategic adjustments to their portfolios.

Short-term directional prediction in volatile markets: We employed NLP-supervised models to predict short-term directional movements in stock prices during periods of high market volatility. This approach can be particularly beneficial for day traders and high-frequency traders, who rely on rapid fluctuations in stock prices to generate profits. By accurately predicting short-term price movements, these traders can execute trades with greater precision and confidence. Incorporating NLP techniques can also enhance the accuracy of stock price predictions by taking into account external factors, such as news articles and social media sentiment. This approach can help investors better understand how market sentiment and external events may influence stock prices, enabling them to make more informed decisions and manage risk more effectively.

Portfolio optimization and risk management: By combining the insights gained from Time Series models and NLP-supervised models, investors and portfolio managers can optimize their investment strategies to maximize returns while minimizing risk. This can be achieved by diversifying investments across various stocks and sectors, adjusting the weight of individual assets based on predicted trends, and implementing risk management techniques, such as stop-loss orders and options strategies.

Algorithmic trading: The predictions generated by our models can be integrated into automated trading systems, allowing for more efficient and precise execution of trades. By leveraging the power of machine learning and AI, algorithmic trading systems can adapt to changing market conditions and capitalize on short-term price movements, potentially generating higher profits and reducing the impact of human error.

Future Development

Our current models, while providing valuable insights, have certain limitations that can be addressed to improve their performance and applicability in various market scenarios.

Incorporating more technical indicators: To enhance the supervised learning model, we plan to include additional technical indicators as features, such as moving averages, MACD (Moving Average Convergence Divergence), RSI (Relative Strength Index), and KDJ (Stochastic Oscillator). These indicators can help capture more nuanced market dynamics, potentially improving the accuracy of our predictions.

Automating the ETL process: In order to streamline the data processing pipeline, we aim to automate the extraction, transformation, and loading (ETL) process. This will allow us to handle large volumes of data, ensuring our models remain up-to-date with the latest market information and trends.

Expanding to other securities and sectors: We plan to deploy our models on stocks and other financial instruments across various sectors, allowing us to compare their performance under different market conditions. This will enable us to identify patterns and trends that are consistent across different industries, as well as those that are unique to specific sectors, enhancing our understanding of market dynamics.

Real-time data integration: Integrating real-time data into our models will help improve their responsiveness to rapid market fluctuations. By incorporating real-time data feeds,

our models can adapt more quickly to changing market conditions, potentially providing more accurate and timely predictions.

Model ensembles and deep learning: Exploring the use of model ensembles and deep learning techniques, such as neural networks, can further improve our prediction accuracy. Combining the strengths of multiple models or leveraging advanced deep learning algorithms can help capture complex market relationships and patterns more effectively.