***Introduction:***

The main aim of this analysis is to observe the trends in the stock market with respect to the news that comes up. We want to know the investor sentiment based on news and how that affects the stock price. In this project, we mainly focus on the collection of stock price data and the news related to it. We consider the Apple company stock which is known by its ticker symbol as AAPL. Now that investing has become widely digital unlike the old traditional methods, the prices of the stock are seeing drastic changes with the positive and negative sentiment of the news. We use the collected news data and perform sentiment analysis using the natural language process tools mainly the VADER analysis. This sentiment score is then combined with the actual stock data. A simple machine learning regression model is built to predict the day’s high based on the sentiment score and other factors such as the open, low, volume, and adjusted close.

***Data Overview:***

The stock price data was collected from Yahoo Finance directly using the “yfinance” package in Python. The financial news data was collected by directly web scraping the FinnHub site using the API key due to its connectivity with Python. FinnHub is a site that provides financial news and other data exclusively related to the stock market and movements. We collected the news which was spread over different time intervals in each day and filtered out the non-English news as we didn’t need it for the analysis using the “LangReview” package in Python. This detects the non-English news and deletes it. After applying the Vader analysis, we were left with the Average Compound score of the sentiment. There are a total of 245 rows and eight feature columns. Details of the final merged dataset are given below:

Table.1: Data Details

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Example Value | Description |
| Date | DateTime | 2023-12-21 | Date of the stock price sample |
| Open | Numerical | 118.4 | Opening price of the stock for the day |
| High | Numerical | 123.2 | Day’s highest price of the stock |
| Low | Numerical | 116.9 | Day’s lowest price of the stock |
| Adjusted Close | Numerical | 119.3 | Adjusted value of the price between various times of closings of the markets |
| Daily Return | Numerical (%) | 0.02% | Calculated field from the adjusted close |
| Volume | Numerical | 12686298 | The number of shares traded that day |
| Average Compound | Numerical | -0.067 | Average compound sentiment score. |

***Exploratory Data Analysis:***

The analysis started with data cleaning even before we got started with exploring the data. The basic statistics of the dataset are given below:

Table.2: Data Statistics

|  | **Open** | **High** | **Low** | **Adjusted Close** | **Volume** | **Daily Return** | **Average\_Compound** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 245.000000 | 245.000000 | 245.000000 | 245.000000 | 2.450000e+02 | 244.000000 | 245.000000 |
| **mean** | 186.448041 | 188.086245 | 184.874735 | 186.140709 | 6.059381e+07 | -0.000735 | 0.090562 |
| **std** | 15.594457 | 15.767412 | 15.228362 | 15.650301 | 2.459044e+07 | 0.013942 | 0.063315 |
| **min** | 165.350006 | 166.399994 | 164.080002 | 164.776505 | 2.404830e+07 | -0.067729 | -0.187162 |
| **25%** | 175.070007 | 176.899994 | 173.539993 | 174.768784 | 4.747140e+07 | -0.007942 | 0.053517 |
| **50%** | 183.419998 | 184.899994 | 181.589996 | 183.131607 | 5.428830e+07 | -0.001550 | 0.094357 |
| **75%** | 192.009995 | 193.000000 | 190.830002 | 191.914474 | 6.606290e+07 | 0.007464 | 0.131502 |
| **max** | 236.479996 | 237.229996 | 233.089996 | 234.820007 | 2.464214e+08 | 0.042598 | 0.301945 |

The above gives information about the important attributes like mean and standard deviation. And here is a depiction of the distributions of the features.

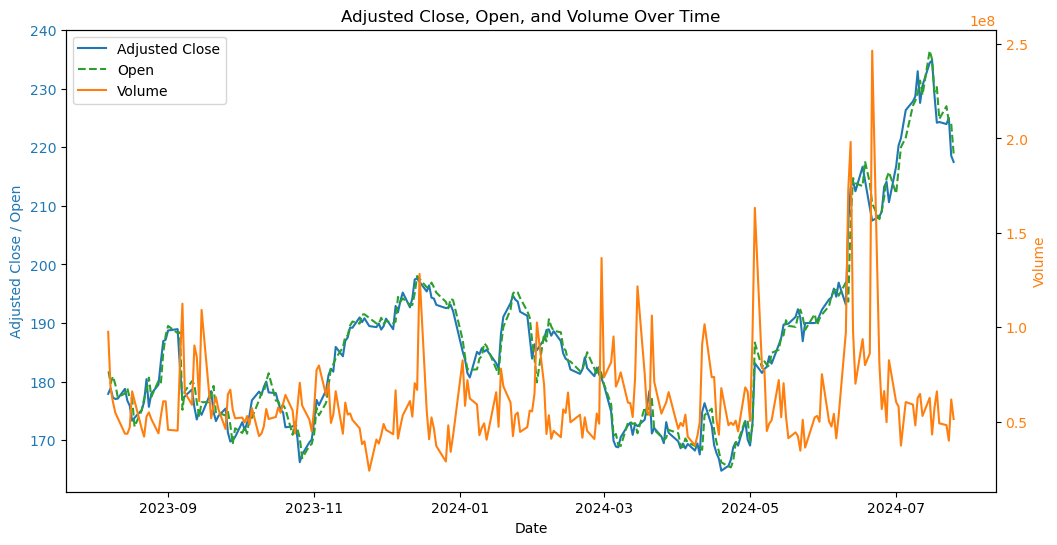


Fig.1. Adjusted close, Open, and Volume over the period

In Fig.1. we can see that there isn’t much difference between the Open and Adjusted close. The volume on the other hand soared just before the adjusted close/ open prices of the stock which is like an indication of the stock growth. From this, we can understand that the high stock prices are a causation of high volume. Volume is nothing but the number of shares trading. We cannot conclude that it is a correlation, but this could be a causation. The causality tests were performed to completely understand this assumption. However, since we have two types of trading: selling and buying the possible reasons for the volume to soar could be selling also, hence we cannot be sure even to consider the assumption of the causation of the high adjusted close and open.

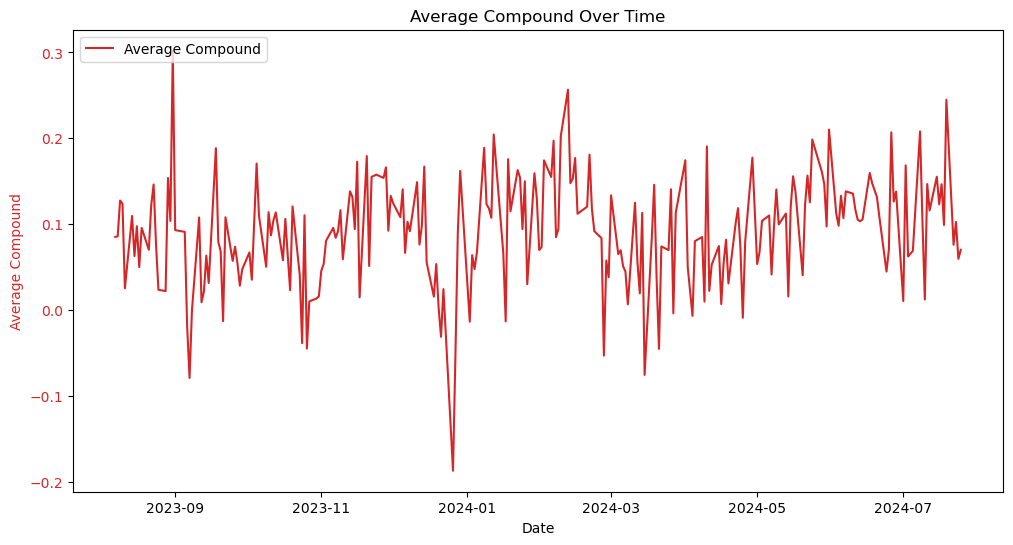


Fig.2. Average Compound over time.

In the Fig.2. we can observe the trend of the sentiment score and understand that it is not necessarily stationary. The compound score shows the correlation with the open/adjusted close prices. As mentioned before, this could be caused by a time gap between the release of the news and the investor's trading time interval. For this purpose, a lag analysis has to be done.

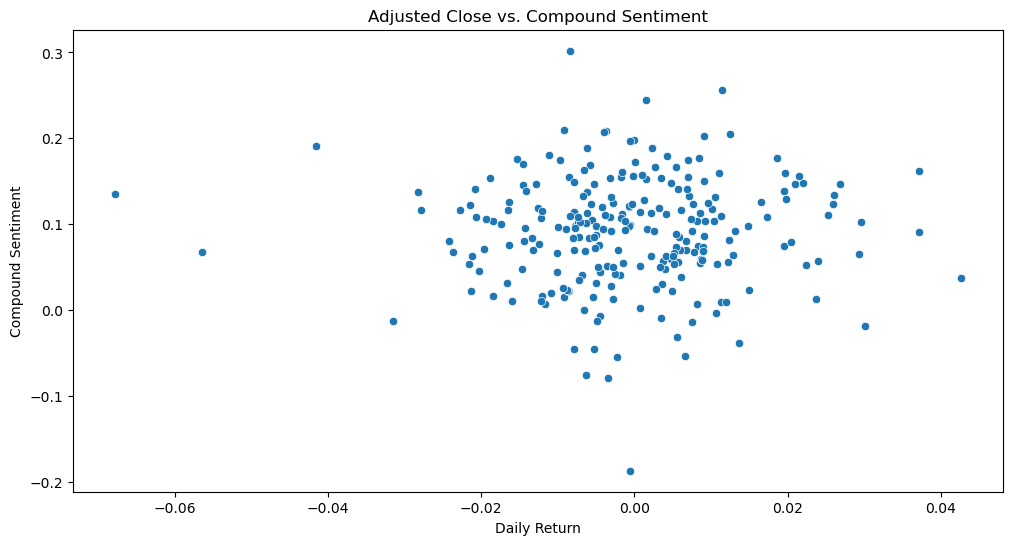


Fig.3. Scatter Plot between Adjusted close and compound sentiment

From the correlation test, the result of the correlation between the adjusted close and the average compound score is close to 0.28. We can observe that from Fig.3. The data points increase from left to right but they are sparse. They are less correlated and positive. However, we considered the “High” as the dependent variable that is to be predicted.

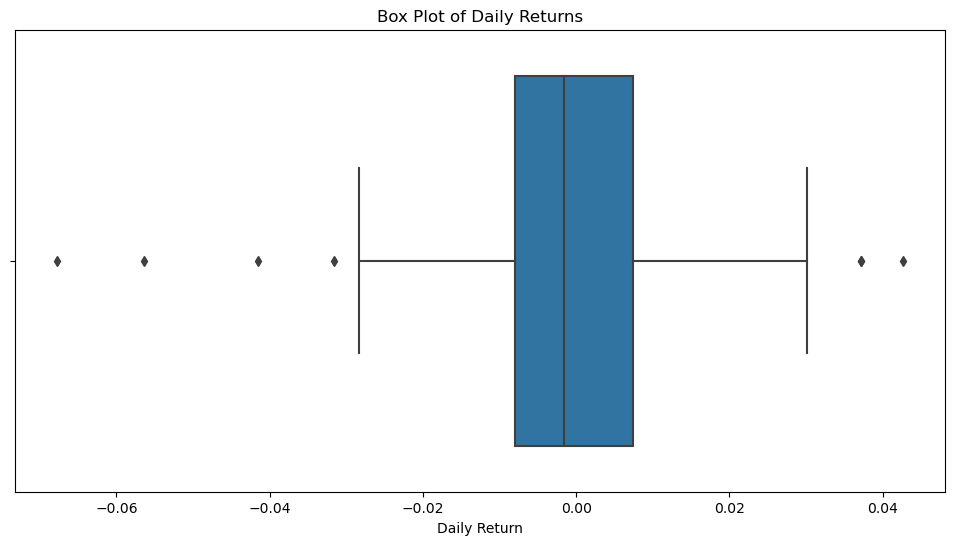


Fig.4. Box & Whisker plot of the daily return

Fig.4. shows the distribution of the data, with the box representing the interquartile range (IQR), which is the middle 50% of the data. The whiskers extend to 1.5 times the IQR from the box, and any points outside of the whiskers are considered outliers. In this particular box plot, the distribution of daily returns is skewed to the right, with a few outliers on the higher end of the return range. This means there were a few days where the stock gained significantly more than usual. The median daily return (the line in the middle of the box) is close to zero. This suggests that, overall, the stock price hasn't changed dramatically over time.

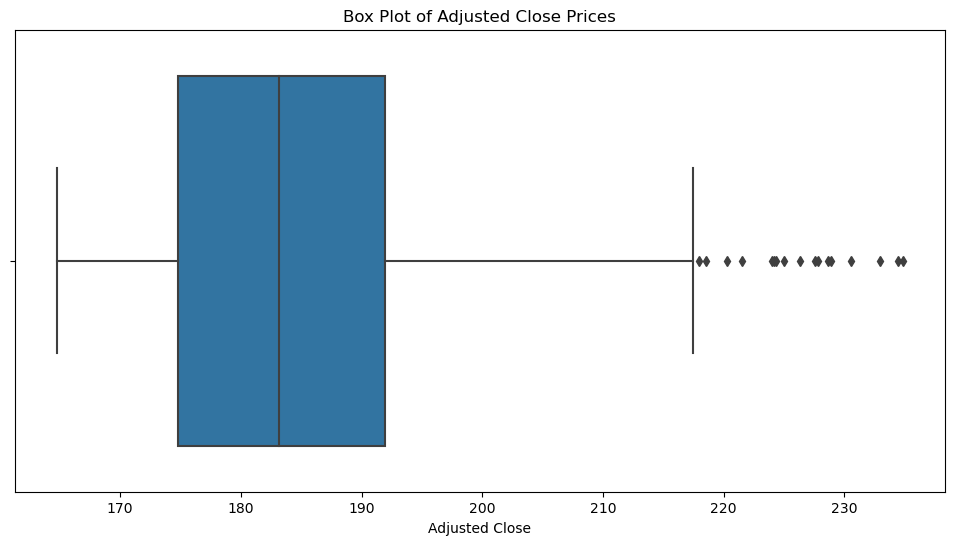


Fig.5. Adjusted Close Prices

Fig.5. plot showing the distribution of adjusted close prices. The box plot shows that the median adjusted close price is around 185, with the majority of prices falling between 175 and 190. However, there are a few outliers with prices above 220.

***Causality Tests:***

The Granger causality test is used to determine the causality between variables. It has two assumptions: Null Hypothesis ( and Alternative Hypothesis (). The Null Hypothesis assumes that the second variable does not granger cause the first variable. Whereas, the Alternative Hypothesis assumes that the second variable granger causes the first variable.

Now in our data, we have multiple features with respect to time, and are all numerical. We aimed to see if the variable “Average\_Compound” sentiment score is causing the highs and lows in the stock price over the time. To perform this test, we used the following library and code in Python :

##code:

from statsmodels.tsa.stattools import grangercausalitytests

grangercausalitytests(df[['Low', 'Average\_Compound']], maxlag=[3])

##end of code

Here are the results of the test for Low values and average compound sentiment scores because we aimed to see if it is caused by Average\_Compound:

Granger Causality

number of lags (no zero) 3

ssr based F test: F=4.9380 , p=0.0024 , df\_denom=234, df\_num=3

ssr based chi2 test: chi2=15.2572 , p=0.0016 , df=3

likelihood ratio test: chi2=14.7937 , p=0.0020 , df=3

parameter F test: F=4.9380 , p=0.0024 , df\_denom=234, df\_num=3

As we can see from the results, the p-value for the F-test=4.9380 is 0.0024, the p-value is less than the significance level of 0.05. Hence the Null Hypothesis is rejected. It’s proven that the average compound score was causing the Low prices.

A similar test was run for the “High” variable as well. And the below are the results for the same:

Granger Causality

number of lags (no zero) 1

ssr based F test: F=6.9987 , p=0.0087 , df\_denom=240, df\_num=1

ssr based chi2 test: chi2=7.0862 , p=0.0078 , df=1

likelihood ratio test: chi2=6.9848 , p=0.0082 , df=1

parameter F test: F=6.9987 , p=0.0087 , df\_denom=240, df\_num=1

The p-value is less than the significance level (alpha) which is 0.0087 and hence the compound score causes the High stock prices.

This was the actual reason why we considered “High” as the dependent variable which needs to be predicted.

***News Data Extraction:***

We used the web scraping method to extract the news data for the AAPL stock prices during a given period of time. There are three different types of web scraping.

1. Traditional Web Scraping: This is where we just do it manually by writing the code to make the API calls and scrape the data as desired
2. Online based applications: There are online based applications which let us use their service. All we have to do is put in our requirements and the url. It returns the data we need.
3. Package based: Some have built in packages in python so that we don’t have to write a separate function to make API calls using the key. We just use those methods/keywords and write a piece of code which is far easier that the traditional we scraping.

We used method 3. FinnHub was used to get the news data. It exclusively gives the financial data related to the company based on the keywords. A class has been defined to carry the attributes like features, the key, start date, end date and ticker. We also used a package called “langdetect” to review the each news sample to detect if there is any non-english news and discard it.

Flow Chart.1: Overview of Webscraping part

HTTP REQUEST

HTTP METHOD CHECK

MERGE AND RUN MODEL

SENTIMENT ANALYSIS - VADER

PRE-PROCESS USING LANGDETECT

FILE DOWNLOADED .CSV

RESULT

API METHOD

AUTHENTICATION

API-PROTECTED

URL CHECK

HTTP RESPONSE

ERROR

MATCHING THE URL

***Sentiment Analysis:***

We used VADER(Valence Aware Dictionary and sEntiment Reasoner) for sentiment analysis. Our data is news which is textual hence VADER just analysis these phrases and gives a score. It is a natural language processing library. The traditional method is to use a set of rules to check for grammatical and syntactical conventions to give a sentiment score. But VADER also uses a sentiment lexicon in combination.

We need to import a package called “SentimentIntensityAnalyzer” from nltk (natural language processing tool kit). VADER gives output as a combination of positive, negative, and neutral scores. We took the compound average of the positive and negative scores as we only needed one score per sample. When feeding data to machine learning models, multiple features have a high positive or negative correlation with each other. Hence our aim is to avoid this as much as possible, which is the reason why we did not include various scores in the final analysis. If the score is greater than 0 then it's positive sentiment. If its greater than or equal to zero then it could be a negative or neutral sentiment respectively.

We did not set any threshold level for this to categorize the samples into groups. However, that score was left as it is as our end goal is to make a regression prediction and not classification. This is not as effective as deep learning models as it struggles with complex sentences. It is mostly favorable with the informal language used in social media posts and articles. For formal news, this is not an option as one might have to consider other better deep-learning options out there. This is great for just a quick and short analysis.

For each day we have many news headlines. As we need only one score per sample we just considered the average of the compound score. This gives us the mean score of the day.

***Machine Learning Algorithms:***

1. SVM Linear Kernel:

Support Vector Machines are used for both classification and regression problems. It is precisely known as Support Vector Regression. It is used when the relationship between the dependent and independent variables is linear. However, when dealing with real-time data, it is never linear. We just assume it's linear and try to get a hyperplane along which the data is predicted. One big problem with this is that it is sensitive to outliers and financial data can have only outliers as the we are not considering all the features that influence the stock price at different time periods. The only problem is that there are no standard features to include in the analysis so that we don’t see any outliers. There are always outliers regardless of what we include. The nature of the stock price movement is volatile.

1. SVM Polynomial Kernel: Unlike linear kernel, poly can handle a bit more complex data and capture the relationships that are usually missed by the linear kernel. It all depends on how we adjust the degree(gamma) value so that we can have control of the model complexity. It projects the data into higher dimensions without transforming it and finds the hyperplane. However, the risk of overfitting increases as we increase the degree to capture the relationships even with small datasets.
2. SVM Sigmoid kernel: This is a special kernel. This gives SVM an opportunity to act as a basic two-layer perceptron making it a neural network without black box nature. However, most datasets have convergence issues with this sigmoid kernel. When implementing this model, a bit of careful fine-tuning is required to achieve decent accuracy, and the decision boundary is hard to understand.
3. SVM rbf: The Radial Basis Function(RBF) kernel has only two main parameters: c (regularization parameter) and gamma. It can handle high dimensional data where we have more number of features than the samples. The choice of values for c and gamma has to be chosen carefully otherwise the model may result in overfitting or underfitting.
4. Linear Regression: It is the most easiest algorithm in the regression problem. It is a benchmark model to run against or compare various datasets and their accuracy with different models. Unlike the above algorithms, linear regression is less prone to overfitting. However, the only problem is this model also works best with the linear data like the SVM linear kernel.

***Difference between SVM(‘linear’) and Linear Regression:***

The primary use of linear regression is in regression problems where we predict a target variable based on one or more independent variables. On the other hand, SVM linear is primarily used for classification tasks but can also be adapted for regression. The objective in linear regression is that the sum of the squared differences between the actual target and the predicted values is to be minimized. Whereas for the SVM linear, the margin between the hyperplane and the closest data points from each class has to be minimized while classifying the data points correctly. Both of them uses different types of cost functions.

***Comparison of the Metrics:***

Table.3: MSE and R^2 values of the algorithms

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SVM(linear) | SVM(poly) | SVM(sigmoid) | SVM(rbf) | Linear Regressor |
| Train MSE | 1.1922 | 44.6066 | 70.0167 | 85.4440 | 1.1459 |
| Test MSE | 2.5399 | 95.8466 | 120.2884 | 139.2604 | 2.3627 |
| Train R^2 | 0.9947 | 0.8026 | 0.6901 | 0.6219 | 0.9949 |
| Test R^2 | 0.9917 | 0.6866 | 0.6067 | 0.5446 | 0.9923 |

As we can see from the above table, SVM(rbf) kernel has the worst train and test MSE(Mean Squared error) values and Linear regressor has the best. SVM(linear) takes the next place as it has values closer to linear regression. The R^2 value of the linear regressor and SVM(linear) indicates that the models were able to catch the pattern in the feature variables and explain the variance in the target variable. This means that these two models were able to predict the stock prices using sentiment score and other variables with a good variation in the prices up to 99%.

***Conclusion and Future Scope:***

In this paper, we approached a basic technique at the end which is to apply simple regression algorithms on the data. However, different emergent techniques like VADER Analysis were used to check the performance of the algorithms with a new Valence method. Most of today’s prediction models include highly tuned neural networks, which are complex and shows their black box nature. We aimed to break down the process into simple steps to make the analysis non-complex yet keeping it accurate and efficient. The nature of the sentiment score of the data has been analyzed and the causality between the main features such as Daily High, Low and the Score has been tested and observed which took us forward to conclude the causation. We can conclude by saying that the simple linear algorithms like Linear regression and SVM(‘linear’) performed better than the complex kernels which use complicated activation functions.

The analysis can be taken forward by experimenting with other algorithms, ensemble learning techniques, and neural networks and picking the best-performing model.

***References:***

1. [Get Financial Data from Yahoo Finance with Python - GeeksforGeeks](https://www.geeksforgeeks.org/get-financial-data-from-yahoo-finance-with-python/)
2. [Financial Data with Finnhub in Python | by Augustin Goudet | Medium](https://medium.com/@augustin.goudet/introduction-to-finnhub-97c2117dd9a9)
3. [How to Perform a Granger-Causality Test in Python (statology.org)](https://www.statology.org/granger-causality-test-in-python/)