**Applications of different forecasting methods  
on stock price data**



Master of Science (MS) in Data Science  
Module: ITC6002A1 – Exploring and Analyzing Data  
Instructor: Dr. Papadias Constantinos

Students’ names & IDs:

Alkiviadis Kariotis 241735

Panagiotis Kounelis 260963

Christos Tryposkiadis 312573

Elisavet Kollarou 275561

Term: Fall Term 2023  
Type: Group Term Final Project – Written Report

Submission Date: Monday 18th of December

Words: 4350

Table of Contents

[Abstract 3](#_Toc153724415)

[Introduction 3](#_Toc153724416)

[Analysis & Discussion 4](#_Toc153724417)

[Dataset Acquisition Description 4](#_Toc153724418)

[Data Preprocessing and cleaning 4](#_Toc153724419)

[Forecasting Methods & Dataset Split 6](#_Toc153724420)

[Naïve Approach 6](#_Toc153724421)

[Moving Average 8](#_Toc153724422)

[Simple Exponential Smoothing 10](#_Toc153724423)

[Summarized results for the three simple methods 12](#_Toc153724424)

[Trend Adjusted Exponential Smoothing Forecast 12](#_Toc153724425)

[Seasonality Adjusted Exponential Smoothing Forecast 16](#_Toc153724426)

[A/F Ratios on Naïve Forecast 19](#_Toc153724427)

[Conclusion 21](#_Toc153724428)

[References 22](#_Toc153724429)

# Abstract

Living through the Big Data era, this paper aims to underline the importance of the simplicity in the forecasting techniques, by providing some insight into the adaptability and the refinement of the simple financial prediction methods. So, this project scope is the analysis of various forecasting time series techniques and the investigation of their effectiveness and performance on future data. Our analysis was performed on stock data obtained from the Yahoo Finance Python library, and more specifically for the Apple Inc for the time period of five years ranging from 2017-2021. We focused on the adjusted closing prices of the stocks both on weekly and on monthly basis, and applied three forecasting methods, namely the Naïve approach, the Simple Moving Average and the Simple Exponential Smoothing. Furthermore, for the Simple Exponential Smoothing, we investigated how the results in terms of MAD and MSE metrics were affected by adjusting trend and seasonality. Finally, the findings indicate the importance of the simplest methods, which sometimes and based on the format of the data being processed, can be proven to be more robust and even outperform complex methods. As for the last part we got our hands on different stock data from other major companies rather in the same industry or similar ones and applied the A/F Ratio approach on our best performing model, which was the Naïve approach. Reaching at a conclusion that by using multiple data and appropriately mixing them together did enhance the results of our forecasting method.

# Introduction

Forecasting is an important method that aims to predict future conditions or data, by utilizing present and past information. It is a very crucial process that businesses utilize in order to make informed future plans and business decisions. Since we live in the age of big data, one important factor to take into account is the need for businesses to make data driven decisions. Past data has to be managed and analyzed using various techniques to predict trends or changes in the future. This can be applied to various business subjects such as allocating resources, strategy planning, financial planning, risk management etc. Especially in the case of companies’ stocks, time series forecasting plays an important role in trying to predict future stock prices based on specific parameters and assumptions. The purpose of this project is to explore and analyze time series data while also employing different forecasting techniques to compare them.

# Analysis & Discussion

## Dataset Acquisition Description

Specifically, the data chosen was the historical stock price data of Apple Inc. stock (AAPL) in weekly intervals. They were downloaded using the Yahoo Finance Python Library. The total time period of our analysis is 5 years, from 01/01/2017 to 31/12/2021. The data was formatted on a weekly basis for the first part of our analysis (Friday) and for the last part on a monthly basis. The initially downloaded data set is separated into 6 distinct columns.

* **Date:** Day for which the stock price data was recorded
* Open: Opening price of the stock on that day
* High: Highest price at which the stock traded during the day
* Low: Lowest price at which the stock traded during the day
* Close: Closing price of the stock on that day
* **Adj Close:** The adjusted closing price of the stock.
* Volume: Trading volume for that day 🡪 total number of shares of the stock traded

## Data Preprocessing and cleaning

For the purposes of our analysis, we only used the date and the adjusted close columns. Specifically, the date column was transformed to a weekly basis because we wanted the data frequency to be weekly. The adjusted close column was chosen as basis since it is the most optimally tuned for statistical analysis. The reason is that the adjusted closing value of a stock price takes into account various different factors such as all applicable stock splits, various dividend distributions and other corporate actions to provide a more accurate reflection of a stock’s price value over time. For example, if we had chosen the simple closing price, which does not incorporate stock splits the historical analysis of the data would be pointless, as the values of the stock through time would be essentially false. In essence, the adjusted closing price is essentially a form of automated data preprocessing. Furthermore, both columns (Date and Adj. Close) where thoroughly checked for the existence of NA values and for duplicates to make sure that all the data is in order. Additionally, the z-score method of detecting outliers was employed, to track any irregularities in the dataset. This technique essentially determines how far from the mean each data point is, measured in standard deviations (Shiffler, 1988). The formula for calculating the z-score is :

Where:

* X is a data point
* μ is the mean of the dataset
* σ is the standard deviation of the dataset

So, using the aforementioned formula the z-score is calculated for every point in the data, and then z-score threshold is chosen to determine which datapoints are going to be considered outliers. Then the actual z-score is compared to the threshold, and if it surpasses it, the data point is considered to be an outlier. In our case the threshold of 3 was chosen which is also the most widely accepted standard for this type of analysis.

The table below displays the descriptive statistics of the weekly historical adj. closing prices of apple from 01/01/2017 to 31/12/2021.

|  |  |
| --- | --- |
| Descriptive Statistics | Adj Close |
| Count | 261 |
| Mean | 72.7346 |
| Standard deviation | 40.84 |
| Min | 27.433 |
| 25% | 40.9292 |
| 50% | 51.7857 |
| 75% | 114.566 |
| max | 177.422 |

The Figure1 below is the stock price chart of Apple Inc from 2017 to 2022:

A graph showing a line graph

Description automatically generated

Figure 1: Apple Inc stock price from 2017-2022

On the y axis we have the adjusted closing price and on the x axis we have the date in years. We can see from the graph that while from 2017 to 2019 the slope is relatively small, from 2019 onwards there is a clear upwards trend with a big increase in the slope. Additionally, from 2019 onwards, we can notice a possible indication of yearly seasonality, as there seems to be a yearly shift in the imaginary trend line.

## Forecasting Methods & Dataset Split

Three different forecasting approaches were implemented on the dataset, the naïve approach, the moving average approach and the exponential smoothing approach. In order to find the best forecasting model based on our data and its parameters we used two key metrics: the Mean Absolute Deviation and the Mean Squared Error. The Mean Absolute Deviation (MAD) is essentially a measure of how much a group of values is dispersed. It displays the degree deviation the data points have compared to the mean on average (Konno & Koshizuka, 2005). The calculation is made by subtracting the mean from each data point in absolute value. The Mean Squared Error determines the average squared differences between the predicted and actual values of a data set. Then we can get the root of the MSE in order to have the units of our dataset. In our case it is dollars since we have stock prices.

To optimize our model parameters, the data set was split into different parts.

* test is the fifth year, only used once to compare the actual 5th values to the forecast.
* train is 1-4th year, used to train our model for the before our 5th year forecast.
* train\_train is 1-3rd year, used to train our model for the before our 4th year forecast (validation).
* train\_validation is the 4th year, used multiple times to compare the actual 4th values to the forecast in order to find the best parameters for each model.

### Naïve Approach

The naïve forecasting approach is arguably the simplest forecasting method for time series analysis. The last available observation is assumed to be the best indicator of future predicted observations (Meyer, 2002). In our case therefore, the last available stock price in the time series is used as a baseline and will be used for the whole forecast. This approach is usually used as a baseline for more complex forecasting methods. The formula is as follows:

Where:

* the most recent observed data, i.e. the last available actual stock price

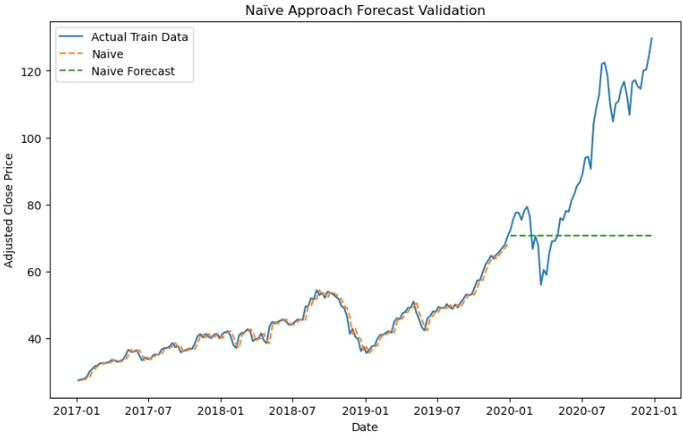
The Figure2 below displays the forecast for the 4th year of the stock prices:

Figure 2: MAD:23.93 & MSE: 927.41

We can see from the graph that the green line is the naïve forecast, and the blue line is the actual data from the beginning of the 4th year onwards. So, the green line keeps the last known historical price steady and applies it throughout the whole forecast. The MAD of 23.93 indicates that on average the predicted stock price deviate from the mean by $23.93 The RMSE (30.45) shows that on average the forecasted values deviate from the actual values by $30.45$

The Figure3 below shows the isolated 5th year forecast is displayed:

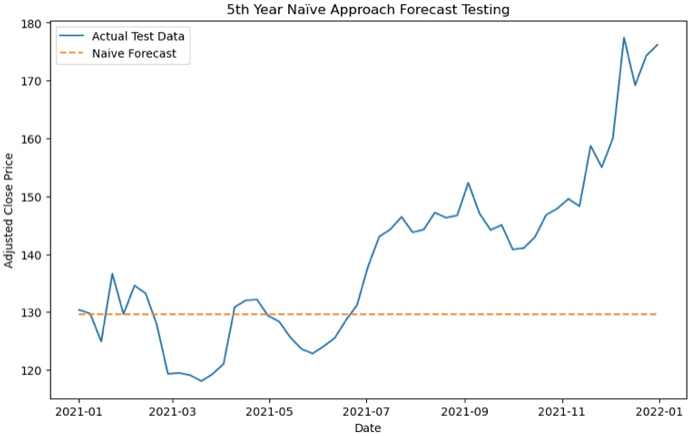


Figure 3: MAD:13.16 & MSE:309.61

Again, as per the naïve approach, the forecast is a straight line based on the last available stock price of the 4th year, which is $130. In this case the MAD and MSE show much better results. The reason is that the during 5th year there was a smaller increase in the slope of the trend of the stock price chart values compared to the 4th year, hence the better accuracy.

### Moving Average

The moving average forecasting method utilizes the average of a specific number of past observations to predict future values. There are a lot of different kinds of moving average but in our case, we employed the simple moving average (SMA) (Johnston et al., 1999). Using python, we assessed the performance of moving average forecasting with various rolling window sizes (from 2 until 156) on the given stock dataset. We visualized and evaluated the accuracy of the different parameters through different evaluation metrics. This provided us with insights on how window size impacts the forecasting performance. When we reached the last available stock price data of the 3rd year (Validation set) and the 4th year (Actual Forecast) we used the first few forecast values depending on the window to predict the following ones. Put simply, when we ran out of actual values to apply the moving average, we used the generated forecasted ones. This is why the moving average forecast ended up being almost a straight line. Specifically, we had forecasted values (FV) and actual values (AV). With a rolling window of 3 for example at the point the forecast starts we had (AV, AV, AV), calculating the average of those we get the first forecast value FV1 (AV, AV, FV1) then calculating the average of those again we get FV2 (AV, FV1, FV2) then again, we get FV3, and we have (FV1, FV2, FV3). Now we don’t have any more actual values and the next forecast values are going to be averaged using the already existing forecasted values hence the forecast line that tends to be straight. The formula is as follows:

Where:

* is the SMA at t time.
* is the observed data point at t-i+1 time?
* the period number in which the rolling average is applied to.

Displayed in the Figure4 is the moving average forecast in the validation set (4th year):

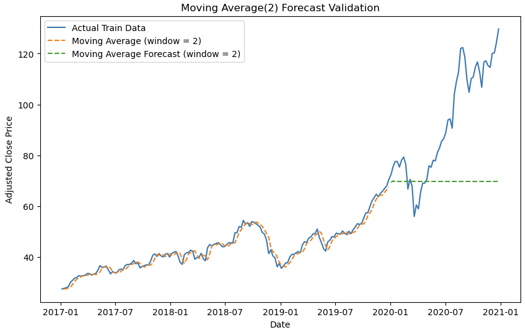


Figure 4: MAD:24.52 & MSE:965.09

Experimenting with different rolling window sizes using loops in python we developed an algorithm that will return the optimal window size based on the minimum values of MAD & MSE. The optimal rolling window return was a size of 2.

Displayed below in Figure5 is the moving average forecast of the isolated 5th year:

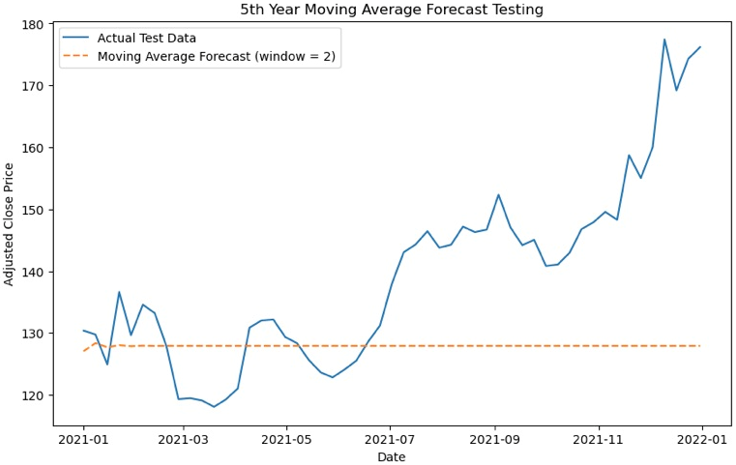


Figure 5: MAD:13.94 & MSE:345.58

Similarly, to the case of the naïve forecast the metrics that resulted from the 5th year forecast are much better compared the 4th for the same reasons. Again, we can see that the naïve forecast for the 5th year slightly outer performs the moving average forecast by a small margin.

### Simple Exponential Smoothing

Like the moving average, the simple exponential smoothing forecasting method uses averages of past values to predict future ones. The distinct difference is that a weighted average is utilized instead of a normal one. A key feature is that the weights are exponentially decreased to past observations (Ostertagova & Ostertag, 2012). The formula is as follows:

Where:

* is the next period’s forecast.
* is the actual value at time t.
* is the forecasted value at time t.
* is the smoothing factor where 0<a<1 and it is used to determine the weight of the most recent value. A smaller alpha increases the weight of older values while larger increases the weight of recent values.

Using python, we assessed the performance of the forecast with various alpha sizes. We visualized and evaluated the accuracy of the different parameters through different evaluation metrics. This provided us with insights on how alpha impacts exponential forecasting performance.

Displayed in Figure6 is the exponential smoothing forecast for the Validation set:

A graph of a graph of a graph

Description automatically generated with medium confidence

Figure 6: MAD:23.95 & MSE:928.53

Using loops in python from 0.01 to 1 with a step of 0.01 in python we identified the optimal alpha that yields the minimum MAD & MSE. The best alpha indicated is 0.99, indicating that there was placed a very heavy emphasis on the weights of the most recent stock prices.

Displayed in the Figure7 is the exponential smoothing for the isolated 5th year:

A graph with a line

Description automatically generated

Figure 7:MAD:13.18 & MSE:310.61

Again, as in the previous models we can see that the 5th year forecast outer performs the 4th year forecast by a significant amount.

### Summarized results for the three simple methods

|  |  |  |  |
| --- | --- | --- | --- |
|  | Naive | Moving Average | Exponential Smoothing |
| MAD | **13.16** | **13.94** | **13.18** |
| MSE | **309.61** | **345.58** | **310.61** |

Without trend and seasonality adjustments, we can observe that the Naïve Forecast method brings back a little bit better results than the other two methods that we tested.

### Trend Adjusted Exponential Smoothing Forecast

Even though from the results that we presented above we say that the Naïve method was just a little bit better than the Exponential Smoothing, we have chosen to investigate the trend adjustment techniques with the Exponential Smoothing. That’s because this technique is found to be particularly useful and bringing back better results when the dataset that it is applied on have trend or seasonality traits. So, by using this more comprehensive forecasting approach, we took into consideration both short term variations and also the overall trend observed in the historical data. Trend Adjusted Exponential Smoothing (which is also known as Holt’s Exponential Smoothing) is designed for forecasting the trend, into time series data, and can be found very useful when the respective data set is showcasing a linear or nearly linear behavior over time (Gardner & Acar, 2019). As, for the implementation part we started by creating two nested for loops for all the possible combinations for alpha and beta. The first loop checked from values ranging from 0.01 to 1 with a step of change at every loop of 0.01. The nested loop for beta, followed exact the same process with the same starting point, step and end point. So, we checked a total of 9801 combinations of alpha and beta which is the result of 99\*99 = 9801. Our model uses the additive trend component, and we searched for the 2 combinations that will bring back the lowest MAD and the lowest MSE respectively.

Displayed in Figure8 and Figure9 is the best combinations for alpha and beta, for the trend adjusted exponential smoothing forecast in the validation set (4th year) together with its plot regarding MAD and MSE correspondingly:

A graph with lines and numbers

Description automatically generated

Figure 8: MAD:7.48 & MSE:87.81

For alpha = 0.01 and beta = 0.53 we achieved the best MAD of 7.48.

A graph of a graph with numbers and lines

Description automatically generated with medium confidence

Figure 9: MAD:7.52 & MSE:86.68

For alpha = 0.03 and beta = 0.77 we achieved the best MSE of 86.68.

Now it was time to apply the model with both of the two best combinations of alpha and beta into the train dataset in order to obtain the forecast of the 5th year:

A graph with a line and a line

Description automatically generated

Figure 10:MAD:25.47 & MSE:786.85

A graph showing the growth of the trend

Description automatically generated with medium confidence

Figure 11: MAD:27.10 & MSE:1280.73

So, based on the results from both combinations for alpha and beta on the forecast of the 5th year of our dataset, we can see that the Figure11 fails to detect the rising trend providing not so good results in terms of MAD and MSE. Additionally, the results with the other combination for alpha and beta shown in Figure10 do not bring back better results. From this we can conclude that although there is a visible trend in our stock dataset, it is very difficult to forecast accurately for a whole year ahead when data is so unpredictable and constantly changing. If we were forecasting only for a week ahead the results would have been better.

### Seasonality Adjusted Exponential Smoothing Forecast

This is a method again used in time series analysis and it is used to make predictions based not only the trend factor adjustment but also on the seasonality aspect of the data. We used that technique, to investigate whether there is a particular repeated pattern in our data in specific time intervals (Souza, Barros, & Miranda, 2007). As, for the implementation part we started by initializing our additive model from python, again on the validation set, exploring a seasonal window of 52 weeks to investigate whether our data have an annual seasonality pattern. Following the same process with the Trend adjusted technique, we created three nested for loops for all the possible combinations for alpha, beta and now gamma. The first loop checked from values ranging from 0.01 to 1 with a step of change at every loop of 0.01. The second nested loop for beta, followed exact the same process with the same starting point, step and end point. And finally, the third most nested loop for gamma, again with the same starting, end point and step. This python code snippet took some hours to be executed because it needed to check 99\*99\*99=970299 different combinations of alpha, beta and gamma and return to us two combinations. One for the three values bringing back the minimum MAD and one more combination for the minimum MSE. Below, we can see the two different three parametric combinations and the respective minimum results that they managed to obtain:

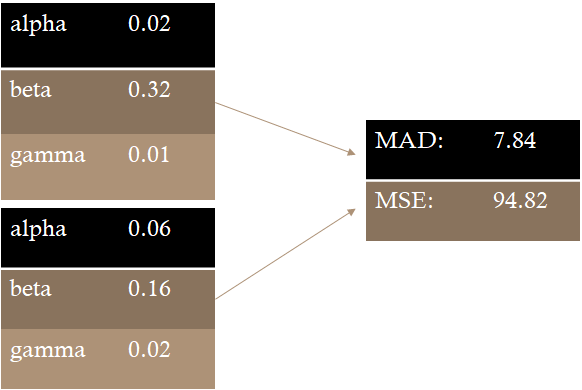


Figure 12: Combinations of alpha, beta and gamma resulting to the best MAD & MSE

And displayed below are both graphs for the two different combinations of Seasonality Adjusted Exponential Smoothing forecast in the validation set (4th year), respectively with the alpha, beta and gamma values combinations shown from the Figure13 & Figure14 and their MAD & MSE are displayed in their caption :

A graph with numbers and lines

Description automatically generated

Figure 13: MAD:7.84 & MSE:100.35

A graph of growth and forecasting

Description automatically generated with medium confidence

Figure 14: MAD:7.99 & MSE:94.82

And finally, the forecast for the isolated 5th year for both of the two three parametric combinations respectively (again with MAD & MSE in the caption of the Figures), are depicted in Figure15 and Figure16:

A graph with orange lines and numbers

Description automatically generated

Figure 15: MAD:22.82 & MSE:654.58

A graph of a line with orange lines

Description automatically generated with medium confidence

Figure 16: MAD:29.16 & MSE:1052.66

So, based on the results from both combinations for alpha, beta and gamma on the forecast of the 5th year of our dataset, we can see from Figure15 does a little bit better than the trend adjusted exponential smoothing forecast but not much better. This means that there is not a lot of seasonality in our data, at least not an annual (52 periods) seasonality. It could be that different seasonality periods value would bring back better results. In order to check what is the optimal seasonality periods value we can use the same logic we used for finding the best alpha, beta and gamma combination, meaning obtain the seasonality periods value that give the minimum MAD and MSE for the 4th year forecasting (validation set). Nevertheless, our results are still worse compared to the simpler techniques. Same reason as the adjusted exponential smoothing forecast, it is very difficult to forecast accurately for a whole year ahead when data is so unpredictable and constantly changing. If we were forecasting only for a week ahead the results would have been better.

## A/F Ratios on Naïve Forecast

Finally, moving to the last part of our analysis, we selected our best performing model, which was the Naïve approach, in terms of MAD and MSE. Firstly, we changed our Apple stock data that we worked on before, from weekly to monthly intervals. Then for the same time period we obtained, again from the python yahoo finance library, similar stock datasets for ten different companies. Here are the 10 companies followed by their ticker symbol:

* Microsoft Corporation - MSFT
* Amazon.com, Inc. - AMZN
* Google (Alphabet Inc.) - GOOGL
* Facebook (Meta Platforms, Inc.) - META
* Tesla, Inc. - TSLA
* Berkshire Hathaway Inc. - BRK-B
* Johnson & Johnson - JNJ
* JPMorgan Chase & Co. - JPM
* Visa Inc. - V
* Coca-Cola - KO

All the data for the ten companies was transformed in monthly intervals. Then, in order to calculate the A/F Ratio of each company, we consider our forecast the second to last day of the 4th year and our actual Adj Closed Price was the last day of the 4th year. Eventually, we proceeded with computing the mean of all the A/F Ratios of all the ten companies (Beaver, 1968). Then we calculated our Naïve Forecast (based exactly on the steps applied previously on the section for the Naïve Forecast) for the 5th year of Apple, but now based on monthly data. Finally, we multiplied the mean obtained from the A/F Ratios with the Naïve forecast for Apple (A/F Adjustment). In Figure17 we can see the Simple Naïve Forecast of the Apple stock data, on monthly basis:

A graph with numbers and lines

Description automatically generated

Figure 17: Simple Naive Forecast with MAD:14.48 & MSE:370.71

The results for the A/F Adjusted Naïve forecast can be seen in Figure18, with the respective MAD and MSE metrics accordingly:

A graph with lines and numbers

Description automatically generated

Figure 18: Naive Forecast of Apple with A/F Adjustment and MAD:13.70 & MSE:290.30

In order to have a better look at exactly the difference we also visualized the Simple Naïve and the Adjusted together in the same plot for the isolated 5th year, which can be shown in Figure19:

A graph with a line

Description automatically generated

Figure 19: Simple & A/F Naive Forecast of monthly Apple stock

So, both MAD and MSE metrics have been better after the A/F Ratio adjustment, implying that there is indeed some kind of relationship between Apple Inc and the other 10 companies we have chosen to analyze.

# Conclusion

Throughout our project, we got a small touch upon the nuance and the complexity of the predictive analytics inside the domain of finance. We demonstrated the difference in terms of application but also of the results that different forecasting methods can project when it comes to the industry of the stock market (historical). Despite its simplicity, the Naïve approach was a surprisingly good method in our context, being able to outperform other more complex techniques when it came to metrics comparison. The insight that we were able to obtain through this occurrence, is that when you are dealing with volatile datasets like the stock prices, maybe sometimes the simplest model can be the most robust one. Nonetheless, valuable insights were provided from the trend and seasonality adjusted models’ analysis. Even though, in this specific dataset the trend- adjusted exponential smoothing did not outperformed the naive method, it was able to highlight the importance of the underlying trends. Likewise, the forecast accuracy for this data set was not amplified from the seasonality adjusted exponential smoothing, which only showed the complexity of identifying seasonal patterns in stock prices and provided us with experience regarding its implementation. Also, an important part of this study was the A/F Ratio that was used on the naive forecast, where data from different important companies were used. The Naïve’s forecasting accuracy of Apple inc. was enhanced by this approach, showcasing that way how it is possible to utilize data from various industries in order to improve the performance of our predictive models. In summary, detailed insights for the stock behavior were offered through the analysis of Apple Inc.’s stock prices, while also outlining principles in financial prediction. We were able to get a glimpse of the opportunities and difficulties of making a prediction about market trends in an economic environment and emphasizes the need of improved forecast methods in order to have more precise and trustworthy financial predictions.

# References

Beaver, W. H. (1968). Market prices, financial ratios, and the prediction of failure. Journal of Accounting Research, 179-192.

Gardner, E. S., & Acar, Y. (2019). Fitting the damped trend method of exponential smoothing. Journal of the Operational Research Society, 70(6), 926-930

Konno, H., & Koshizuka, T. (2005). Mean-absolute deviation model. IIE Transactions, 37(10), 893-900

Meyer, D. (2002). Naive time series forecasting methods. R News, 2(2), 7-10

ohnston, F. R., Boyland, J. E., Meadows, M., & Shale, E. (1999). Some properties of a simple moving average when applied to forecasting a time series. Journal of the Operational Research Society, 50(12), 1267-1271

Ostertagova, E., & Ostertag, O. (2012). Forecasting using simple exponential smoothing method. Acta Electrotechnica et Informatica, 12(3), 62

Shiffler, R. E. (1988). Maximum Z scores and outliers. The American Statistician, 42(1), 79-80

Souza, R. C., Barros, M., & Miranda, C. V. C. (2007). Short term load forecasting using double seasonal exponential smoothing and interventions to account for holidays and temperature effects. In TLAIO II-2 do Taller Latino Iberoamericano de Investigación de Operaciones (pp. 1-8). Acapulco, México.