Capstone Project 2
Apple Stock Price Prediction

Elinor Storkaas

Table of Contents

Background

In my project I decided to analyze the Apple Stock (AAPL) price from the years 2015 to 2022. The data is collected from Yahoo Finance.

Inside the data set there are 7 different variables: 6 numeric and 1 categorical. Each of these variables are correlated to the stock, and I will look more detailed into the open and closing price through the years.

Once I have done an analysis of the variables through the years, I will look in more detail about the prediction of the future stock price. The Data set is divided in 7 columns with 1994 rows.

Column Explanation

Date	The date of the stock price.
Open	The price at which the financial security opens in the market when trading begins.
High	The highest price at which the security has traded during the current trading day.
Low	The lowest price that a stock trades in that day.
Close	The last price at which a stock trades during a regular trading session.

Adj Close	The closing price after adjustments for all applicable splits and dividend distributions
Volume	The number of shares traded in a particular stock, index, or other investment over a specific period of time.

Problem Scenario

Apple Inc. is a global American technology firm with headquarters in Cupertino, California. Apple is the world's largest firm by market capitalization as of June 2022, the fourth-largest personal computer vendor by unit sales, the second-largest producer of mobile phones, and the largest technological business by revenue (totaling US$365.8 billion in 2021). Together with Alphabet, Amazon, Microsoft, and Meta, it is one of the Big Five American IT firms.
In my project I would like to predict the stock price of Apple. As you can already see in the columns I will be using, I will analyze the stock price from all the variables.
I will mainly use exponentially weighted moving average(EWMA), simple moving average(SMA) and the SARIMA model for my prediction.

Objective

- Filter and understand the dataset
- Create a prediction model for the stock

- Evaluate the different models and choose the best one for prediction

Data Exploration

I downloaded the data set from Yahoo finance and uploaded the CSV file in my python code.

```
import pandas as pd
import matplotlib.pyplot as plt
apple = pd.read_csv("AAPL.csv")
```

I will have 7 distinct columns and 1994 rows, so I needed to make sure there won't be an excessive number of NAs before I started. Having missing values when developing a prediction model may result in an incorrect interpretation of the data and a prediction that is not particularly accurate.
To prevent this, I have decided to consider removing a column if it contains an excessive number of NAs.
apple.isna().sum()

Fortunately, the dataset doesn't contain any NAs, as I discovered after running the code, so I can move forward with the further exploration of the data.

I also wanted to check the info of the data set do see the data types of each variable.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1994 entries, 0 to 1993
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       1994 non-null   object
 1   Open       1994 non-null   float64
 2   High       1994 non-null   float64
 3   Low        1994 non-null   float64
 4   Close      1994 non-null   float64
 5   Adj Close  1994 non-null   float64
 6   Volume     1994 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 109.2+ KB
```

After checking there all the columns are numeric except for the 'Date' column, so I needed to convert it into datetime object.

df['Date']=pd.to_datetime(df['Date'])

	Open	High	Low	Close	Adj Close	Volume
count	1994.000000	1994.000000	1994.000000	1994.000000	1994.000000	1.994000e+03
mean	72.928725	73.764658	72.120697	72.974867	71.327663	1.315391e+08
std	48.914185	49.566897	48.268471	48.939470	49.458508	6.774176e+07
min	22.500000	22.917500	22.367500	22.584999	20.914917	3.519590e+07
25%	32.500000	32.689375	32.225624	32.496251	29.499118	8.642390e+07
50%	48.039999	48.582500	47.774999	48.165001	46.369884	1.120656e+08
75%	123.287502	124.797498	121.245001	122.985000	121.681018	1.556468e+08
max	182.630005	182.940002	179.119995	182.009995	180.959732	6.488252e+08

I reasoned that having a summary and description of each column could also be helpful to me. The count, mean, standard deviation, lowest and maximum, 25%, 50%, and 75% percentiles are all displayed in the visualization below.

Data Visualization

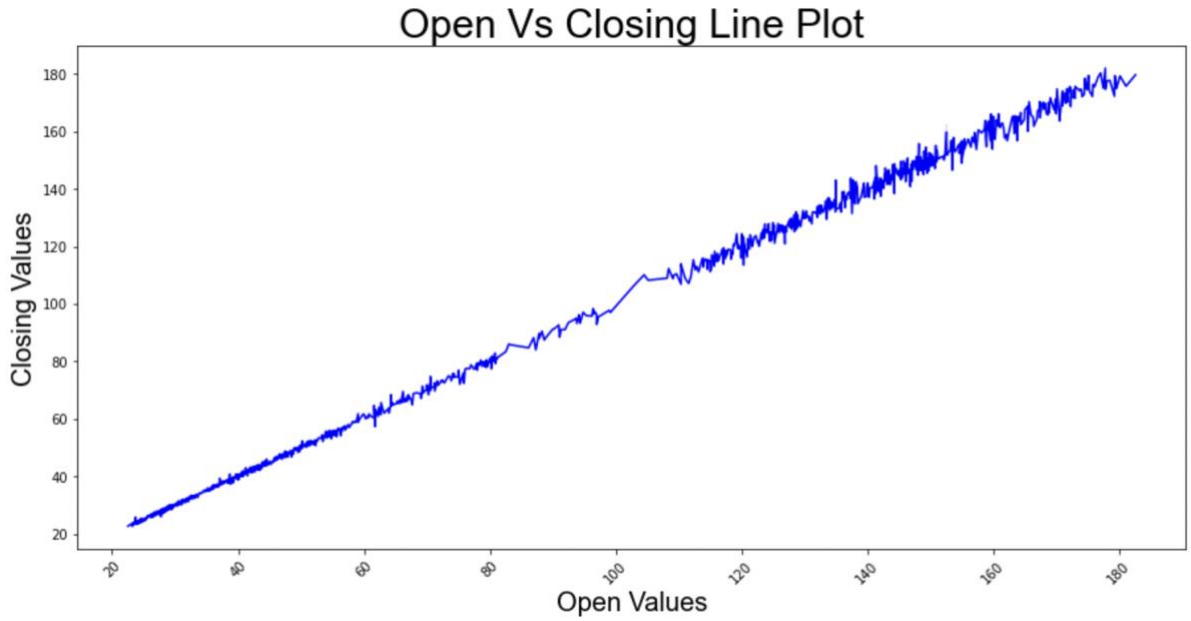
For the data visualization I wanted to find the top 10 open values by date and the 10 highest closing values by date.

	Date	Open
1754	2022-01-04	182.630005
1739	2021-12-13	181.119995
1749	2021-12-28	180.160004
1755	2022-01-05	179.610001
1751	2021-12-30	179.470001
1750	2021-12-29	179.330002
1742	2021-12-16	179.279999
1813	2022-03-30	178.550003
1752	2021-12-31	178.089996
1814	2022-03-31	177.839996

	Date	Close
1753	2022-01-03	182.009995
1748	2021-12-27	180.330002
1754	2022-01-04	179.699997
1738	2021-12-10	179.449997
1750	2021-12-29	179.380005
1741	2021-12-15	179.300003
1749	2021-12-28	179.289993
1812	2022-03-29	178.960007
1816	2022-04-04	178.440002
1751	2021-12-30	178.199997

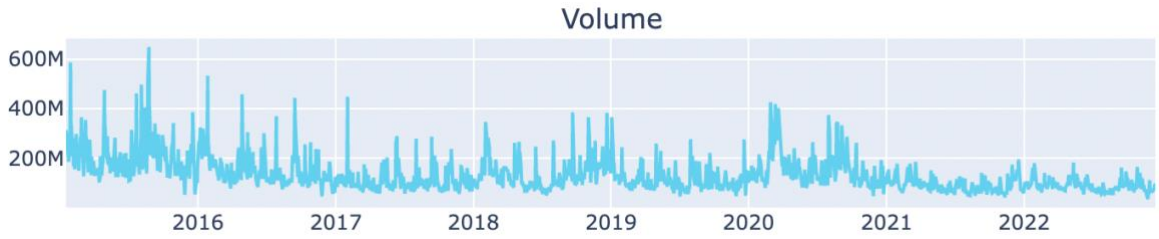
The highest opening date was 4th of January 2022, and the highest closing date was 3rd of January 2022. The lowest opening value was on May 13th, 2016, and the lowest closing value was on May 12th 2016.

I made a line plot chart of the correlation between the closing and the opening values.



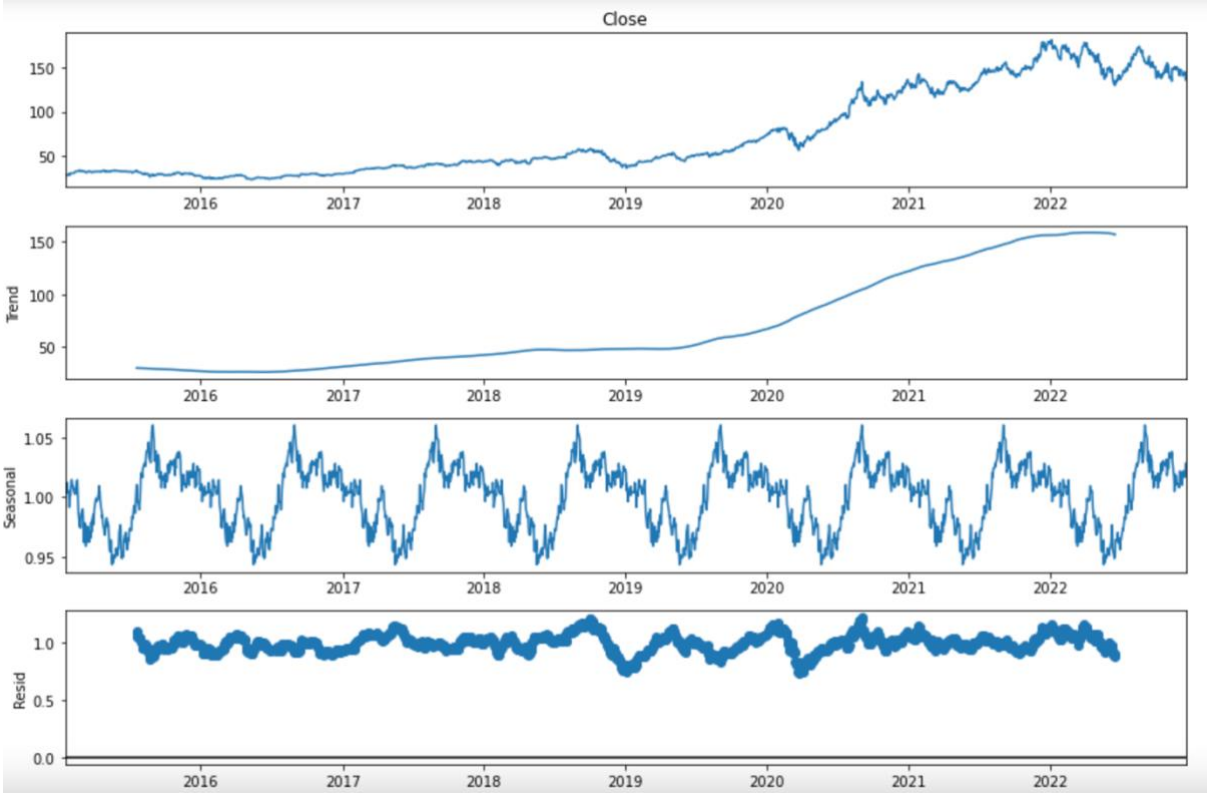
We can see that the open and closing values follow eachother in value.

To get a better understatning of the different variables I wanted to create a time series analysis of each variable with a line chart.



Seasonal Decomposition

The seasonal decomposition can be used to analyze time series that are impacted by variables that cycle (periodically) change over time. When I enter 252 into the parameter "period," it is clear that there is yearly seasonality because there are typically 252 trading days every year. I did it for the closing values.



To predict the future stock price I had to train some of the data. Since SARIMA will be the model I employ, training it with a seasonal length of 252 takes some time. Passing in four for quarterly data or twelve for monthly data is typical. To give the data a monthly periodicity, I therefore total the closing prices for a given month. I take 80% as training set and 20% as test sets.

```
apple = apple.reset_index()

size = int(len(apple)*0.8)
train = apple.loc[:size,["Date", "Close"]]
test = apple.loc[size+1:, ["Date", "Close"]]

apple = apple.set_index("Date")
train = train.set_index("Date")
test = test.set_index("Date")
```

I created a function to run the data through the Augmented Dickey-Fuller Test. The ADF Test examines the alternative hypothesis that the time series data are non-stationary because they lack a unit root.

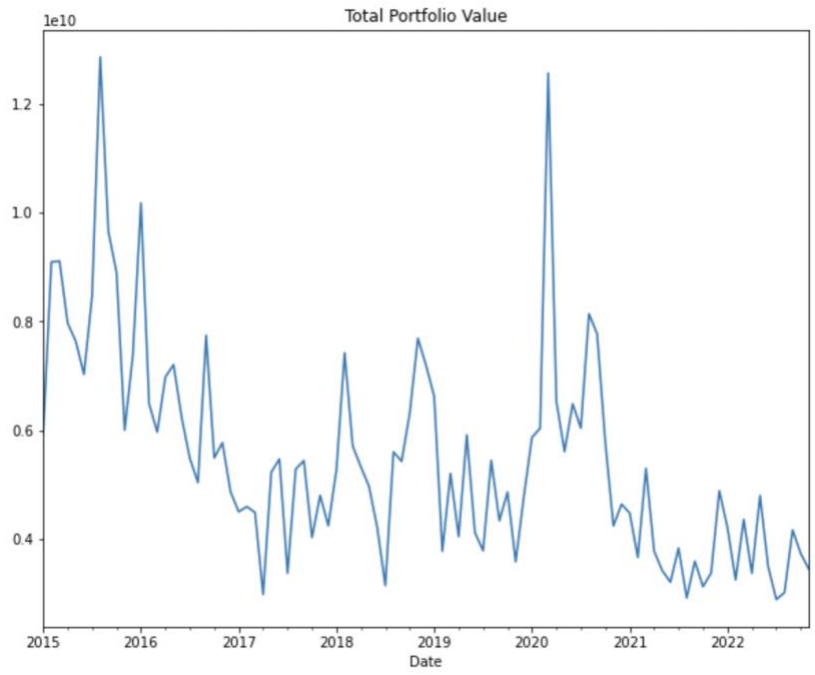
```
def adf_test(data):
    result = adfuller(data)
    print(f'ADF Test Statistic: {result[0]}')
    print(f'p-value: {result[1]}')
    print(f'Number of Lags: {result[2]}')
    print(f'Number of Observations Used: {result[3]}')
    print('Critial Values:')
    for key, value in result[4].items():
        print(f'\t{key}, {value}')
```

You can see that I am unable to rule out the H0 (null) hypothesis. The data must then be differentiable so that time series data is steady and has no unit root. Indicating that the data's mean and variance remain stable throughout time.

I have enough data after performing the ADF test on the differenced data to reject the null hypothesis because the p value is far too low compared to 0.05. Here, I have observed that a one-step difference in the data is sufficient to make the data stable, therefore I choose a "d" value of 1.

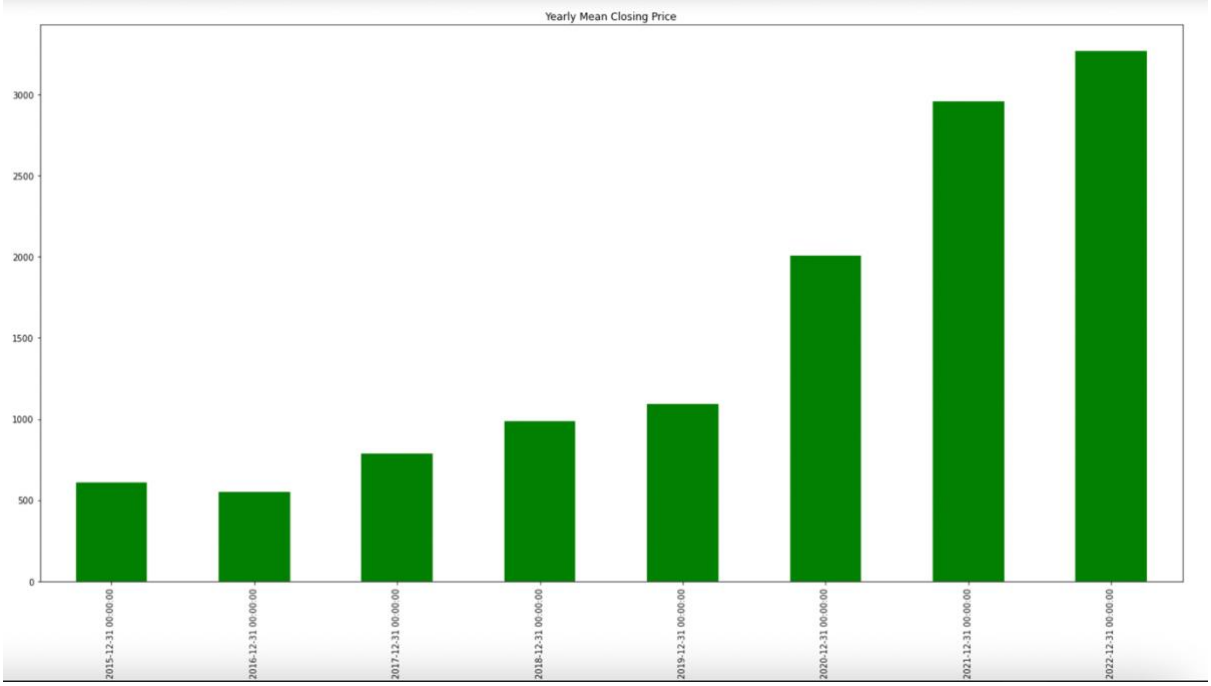
Data Manipulation

I wanted to find the total portfolio value of the stock by adding a new column to the data frame. portfolio value" means the total monetary value of the assets held in your investment portfolio.



We can see that the highest portfolio value was half through 2015 and in the beginning of 2020.

I also created a bar chart of the average closing price:

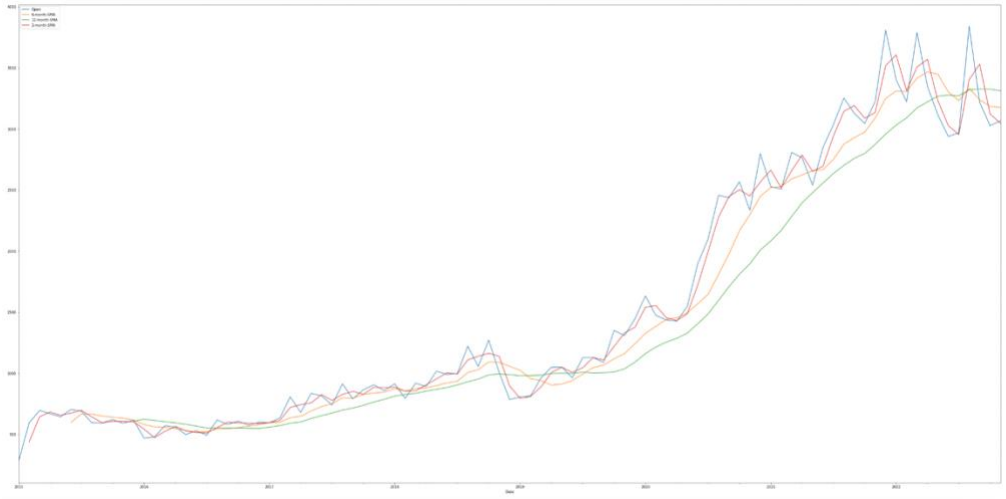


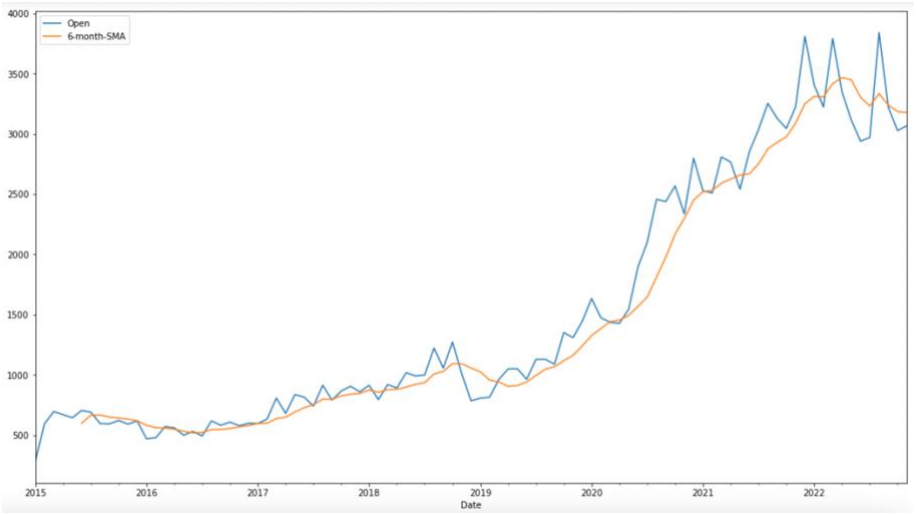
The highest average closing price have been increasing since 2015, which means that Apple has increased in value.

Model Building

For the model building I wanted to train my dataset on the AdaBoost algorithm. The abbreviation "AdaBoost" stands for "Adaptive Boosting," which is a very well-liked boosting approach that turns several "weak classifiers" into one "strong classifier." This will better the performance of the random forest score I did next. The random forest score is 0.9999834420573994 and the random forest model's test score is 0.9998654209595209. This is a very high score which is good, and it means that the model is very accurate. Next, I checked the mean squared error of the AdaBoost model is 2487. This value is high which means there is a big difference between actual and predicted values.

Furthermore, I wanted to find the simple moving average (SMA) for prediction. It calculates the average stock price over a predetermined number of periods in each range. I calculated the 6-, 12- and 2-month SMA.

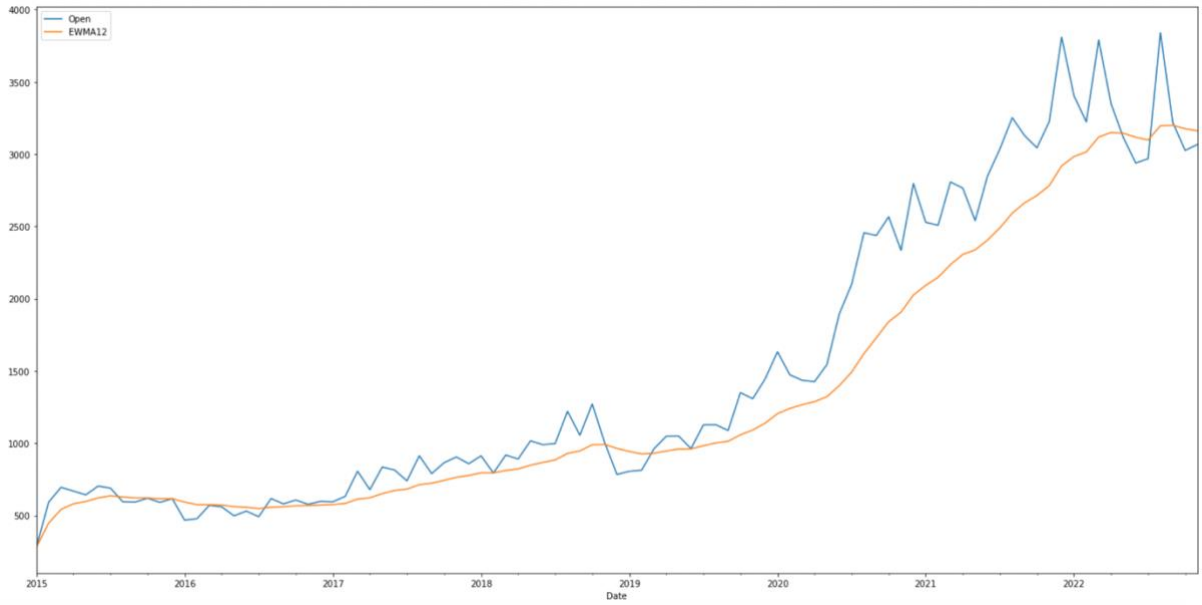




I also wanted to find the exponentially weighted moving average (EWMA) of the open price of the stock. The idea behind employing a moving average is to give newer data points more weight and less weight to older ones. With increasing age of the data points, the weights

decrease exponentially. A simple moving average that gives each data point in each time period the same weight responds to recent process changes more slowly than an exponentially weighted moving average.

This chart shows the EWMA compared to the actual opening price.



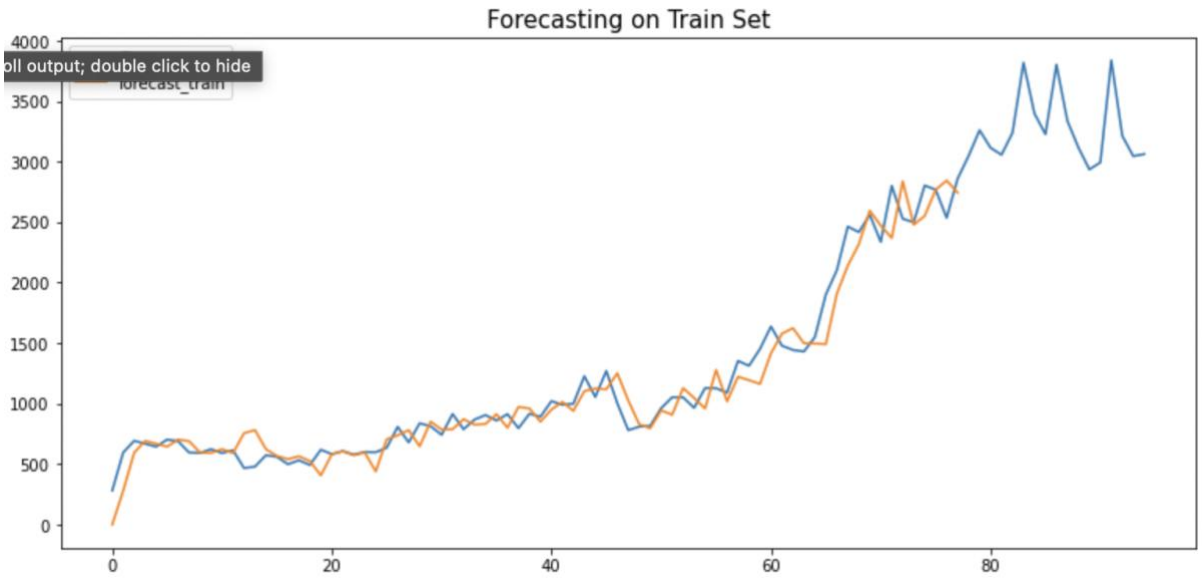
We can see that the SMA is more accurate for the actual opening price than the EWMA.

SARIMA – model

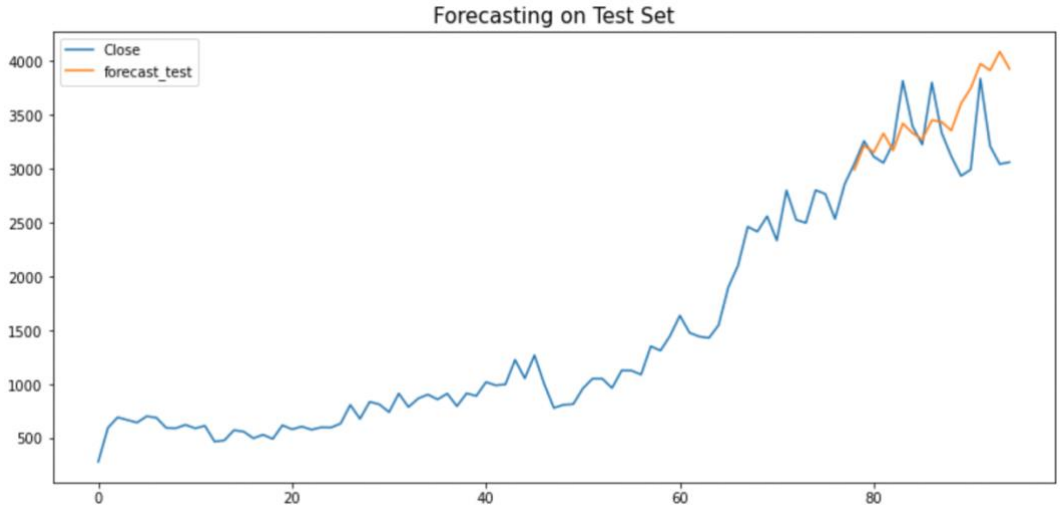
I wanted to use a SARIMA model for further predictions about the closing price. Since the data comprises the monthly sum of closing prices, I have removed the month of December 2022 from the calculation because we are in December 2022, and we can only calculate the monthly sum of closing prices in January 2023. (Which is future).

```
apple.drop("2022-12-01", axis=0, inplace=True)
apple = apple.reset_index()
```

The forecasting on the train set is as follows:



The forecasting on the test set is as follows:



The future forecasting into 2023 is as follows:

As we can see the future closing price is predicted to rise into 2032.

Model Selection

Trying to select the best model for my prediction is not easy. The AdaBoost model made a very inaccurate model which made the performance on the random forest score worse. I made a simple moving average and an exponentially weighted moving average for the opening price of the stock. From the graph we could see that the SMA performed better since it was closer to the actual price. Then, I made a SARIMA model to predict the future price. I used the previous training and test sets to perform this model. The forecasting on the trained set is more accurate than the test set but it still performs quite good in the future prediction. I would choose the SMA and the SARIMA models as the best models for my predictions.

Conclusion

I wanted to analyze a stock of a well-known company so I could be able to relate to it as much as possible. That is why I chose Apple, because they have had an amazing journey since they started. However, I wanted to take the most recent data because I wanted to focus

on how the company is performing in newer times, and into the future. The data I have analyzed is from year 2015 to 2022.

I explored the data set to get a sense of how it was and prepare it for further exploration and manipulation. There are seven different variables in the data set: Date, Open, High, Low, Close, Adjusted Close, and Volume. I found the highest closing price which was on 4th of January 2022 and made a line chart of the correlation between closing and opening price of the whole data set. The values correlate well with each other.

I trained the model to use it in the SARIMA model. The model performed well and predicted a rising closing price in the future. The model is of course not 100% accurate so we don't know for sure that the closing price will continue to rise into the new year, but it is likely to.

Works Cited

https://finance.yahoo.com/quote/AAPL/history?p=AAPL

https://www.kaggle.com/datasets/tarunpaparaju/apple-aapl-historical-stock-data

https://en.wikipedia.org/wiki/Apple_Inc.