Elona Zharri

Final Report
Stock Forecasting

Problem Statement

Stock market is one of the major fields that investors are dedicated to, thus stock market price trend prediction is always a hot topic for researchers from both financial and technical domains. Modeling the markets using advanced financial engineering techniques has lately attracted a great deal of attention. There have been many attempts to predict the behavior of bonds, currencies, stocks, stock markets or other economic markets. These attempts were encouraged by various pieces of evidence that economic markets do not behave randomly, but rather perform in a chaotic manner (Malliaris *et al.*, 1994)[1].

In the earlier stage, under the assumption of efficient market investors believed that the movement of stock price presents a state of random walk. That means it is impossible to predict the change of stock price by its historical data. Nevertheless, some researchers who did empirical studies applying investment portfolios found historical information is actually useful in prediction.

The domain of financial time series data is complicated due to its random walk process, unpredictable day to day variations and being time dependable. Due to the fact that stock markets are affected by many highly interrelated economic, political and even psychological factors, it is very difficult to forecast the movement of the stock market.

This project is focused on answering “How close can we get in predicting the stock market.”, by utilizing a variety of exploratory techniques and different supervised machine learning algorithms such as: Linear Regression, Random Forest, Support Vector Machine and Gradient Boosting Trees.

Data

For this project I have used the Apple Dataset ranging from 1996 to 2020, consisting of the following fields: **Date**, **Open**, **Low**, **High**, **Close**, **Adjusted Close** and **Volume**. Some features of the dataset were classified as weakly correlated with the prediction, therefore were reduced. Additional mathematical calculations were performed to generate technical indicators as new features for the dataset, as technical indicators are strong pattern-based signals produced by price, volume and/or open interest, used to predict the price movements. The technical indicators generated for this project are: Moving Average, Moving Average Convergence Divergence, Relative Strength Index and Bollinger Band. These specific indicators were selected, as they are key indicators used for technical analysis to provide an estimation of the data trend.

Data Wrangling

The raw dataset consisted of 6193 rows and 7 columns. The size of the dataset was normal, and there was no need in reducing the records.

The 'Close' column was dropped, and the 'Adjusted Close' column was renamed to 'Close', as it will give a better idea of the overall value of the stock and helps us make informed decisions about buying and selling, while the closing stock price will tell you the exact cash value of a share of stock at the end of the trading day. Later on the Feature Engineering step, other features such as (Open, Low, High, Volume) were reduced as well and additional technical indicators features were added.

The dataset was robust, and no missing records were found.

After loading the dataset, the Apple data frame consisted of float, integer and object data types. Since time series data can come in different formats, a required adjustment is to convert the 'Date' column from object into a datetime data type. Changing the 'Date' column into DatetimeIndex and assigning it as the index of the dataframe, provides better flexibility in resampling data.
The data frame was sorted based on the 'Date' index, and the final shape of the data frame was (6193,6).

Exploratory Data Analysis

When working with time series data, an important step is to check for data's stationarity. Stationarity is important because many useful analytical tools and statistical tests and models rely on it.

The most basic methods for stationarity detection rely on plotting the data, or functions of it, and determining visually whether they present some known property of stationary (or non-stationary) data.
To determine the data stationarity I plotted the moving average and moving variance and see if it varies with time. By moving average/variance I mean that at any instant 't', we'll take the average/variance of the last year, i.e. last 12 months. This visualization illustrated an increasing trend in the data.

I used the DickeyFuller Test, as a statistical test to check stationarity. The test results consist of a Test Statistic and some Critical Values for different confidence levels. If the 'Test Statistic' is less than the 'Critical Value', we can reject the null hypothesis and say that the series is stationary. In our case the 'Test Statistic' was higher than the Critical values, indicating that the data is not stationary, so we can not reject the null hypothesis that the series is a random walk.

There was no seasonality on the dataset, but there was a trend. One of the first tricks to reduce trends was transformation. For example, in this case there is an exponential growth and significant positive trend. So I applied transformations which penalize higher values more than smaller values by taking the log and the difference of the series. By taking only the log of the series we simply eliminate the exponential growth. Applying the difference and the log at the same time, the transformed series looks stationary.
This approach did improve the ADF test results, but a drawback in this particular approach is that the time-period has to be strictly defined. In this case I took the 'weighted moving average' where more

recent values are given a higher weight. There are many techniques for assigning weights. A popular one is the exponentially weighted moving average where weights are assigned to all the previous values with a decay factor, which I have used.

This approach did generate good results, with a 'Test Statistic' value of -9.936328e+00 and 'Critical Value (1%)' of -3.431410e+00, rejecting the Null Hypothesis that the series is a random walk, and establishing the stationarity of the data.
We update the 'Close' column with stationarized data. After eliminating the trend, the next step was calculating the technical indicators and updating the dataset with the new features.

Feature Engineering

Moving Average Technical Indicator: The reason for calculating the moving average of a stock is to help smooth out the price data over a specified period of time by creating a constantly updated average price. A simple moving average (SMA) is a calculation that takes the arithmetic mean of a given set of prices over the specific number of days in the past; for example, over the previous 15, 30, 100, or 200 days. Exponential moving averages (EMA) is a weighted average that gives greater importance to the price of a stock in more recent days, making it an indicator that is more responsive to new information.[2]
The function which calculates the MA for this dataset, provides different parameters to adjust the short and long window, the type of moving average (SMA, EMA), and also the starting-ending date.
I used the SMA for 20 and 50 days, and generated two additional columns for the dataset, 20_SMA and 50_SMA.

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

where:

$A =$ Average in period n

$n =$ Number of time periods

Moving Average Convergence Divergence Technical Indicator: Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA.
The reason for calculating the MACD is to trigger technical signals when it crosses above (to buy) or below (to sell) its signal line. The speed of crossovers is also taken as a signal that a market is overbought or oversold.MACD helps to understand whether the bullish or bearish movement in the price is strengthening or weakening.[2]

$$\text{MACD} = \text{12-Period EMA} \ - \ \text{26-Period EMA}$$

With the MACD function I calculated the MACD and MACD Signal, which were added to the dataframe as additional features.

Relative Strength Index Technical Indicator: The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100. An asset is usually considered overbought when the RSI is above 70% and oversold when it is below 30%.

$$RSI_{\text{step one}} = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}}\right]$$

The average gain or loss used in the calculation is the average percentage gain or loss during a look-back period. The formula uses a positive value for the average loss.[2]
The standard is to use 14 periods to calculate the initial RSI value, so I also used a lookback period of 14, and calculated the RSI_14 for the dataset.

Bollinger Band Technical Indicator: Is used to generate oversold or overbought signals.
There are three lines that compose Bollinger Bands: A simple moving average (middle band) and an upper and lower band. The upper and lower bands are typically 2 standard deviations +/- from a 20-day simple moving average, but can be modified.[2]

$$\text{BOLU} = \text{MA}(\text{TP}, n) + m * \sigma[\text{TP}, n]$$
$$\text{BOLD} = \text{MA}(\text{TP}, n) - m * \sigma[\text{TP}, n]$$
where:
$\text{BOLU} = \text{Upper Bollinger Band}$
$\text{BOLD} = \text{Lower Bollinger Band}$
$\text{MA} = \text{Moving average}$
$\text{TP (typical price)} = (\text{High} + \text{Low} + \text{Close}) \div 3$
$n = \text{Number of days in smoothing period (typically 20)}$
$m = \text{Number of standard deviations (typically 2)}$
$\sigma[\text{TP}, n] = \text{Standard Deviation over last } n \text{ periods of TP}$

I used the 20_SMA column to generate the Upper Bollinger Band and Lower Bollinger Band, and added these additional columns to the dataset.
After finalizing all the indicators, and adding them on the dataset, I also dropped ‘Open’, ‘Low’, ‘High’, and ‘Volume’ columns, as none of them will be used for the prediction.
The final shape of the dataset is now (1693,8), with Date as its index.

Scaling and splitting the data

Initially I randomly split the data, by using the *train_test_split()* method, but we realized that the model was producing a perfect prediction, which was somehow dubious.

Therefore instead of splitting the data randomly, I used the *TimeSeriesSplit()* to split the data for training and testing into three different sets.

Build Models and Make Predictions

For this dataset I decided to use 4 supervised machine learning algorithms, to build different models and compare the predictions. I used Linear Regression, Random Forest, Support Vector Machine and Gradient Boosting Trees.
The predictions did differ from model to model, although the change was small.

Model Selection

After comparing all the models, I chose to select the Random Forest model as the best fit for this dataset, based on the mean squared error, which was lower compared to others.
I used RandomizedSearchCV and GridSearchCV to hypertune the model parameters, which improved the performance by 4.44%.

Future Research

With this project I was able to better understand the time series data, and I am interested in testing these models with different stock market data, and observing the behavior. A good stock indicator could be Tesla or Ford, as these datasets might also have seasonality and trends. An additional future development would be to feed the dataset into an Arima model and analyze it's predictions.

References

1. Modeling the behavior of the S&P 500 index: a neural network approach, Malliaris *et al.*, 1994, https://www.researchgate.net/publication/3559429_Modeling_the_behavior_of_the_SP_500_index_a_neural_network_approach

2. Technical Analysis, https://www.investopedia.com/terms/t/technical-analysis-of-stocks-and-trends.asp