CS 365 - Final Report

Aditya Chowdhri

April 2024

1 Introduction

Money continues to play an important role in everyone's life, and being able to predict the movement of the stock market with any degree of accuracy provides many financial opportunities.
Traditional approaches used by quantitative trading firms fall primarily under two main categories. The first uses "technical indicators" which are certain stock movement patterns observed through candlestick graphs. The second is fundamental analysis" which aims to use information about the company and its intrinsic value to make purchasing decisions.
An emerging method is to train a machine learning model and predict future prices. My report is driven by three primary questions: 1. *How can feature engineering help fine tune machine learning models to improve accuracy.* 2. *How accurate can models be created and what returns can be expected?.* 3. *Which machine learning algorithms work well with time-series data?*
While these questions will guide my study, I will try to address other questions that arise during the project adequately. With these ideas in mind, I would like to address which stock I will be using. Although the goal of the report is to create a generalized algorithm, diagnosing specific issues is made easier when considering a well-recognized and researched company. For this reason, I will be using Apple's stock [NASDAQ: AAPL]. Personally, I am knowledgeable of the company's workings, making it easier for me to contextualize the work I produce.

The overarching goal of this project is to accurately predict future stock prices based on historical prices and related features.

2 Literature Review

Time-series data is often difficult to work with, and therefore, past literature needs to be consulted with regard to appropriate methodology and feature engineering. Parts of the scikit learn documentation for time-series data proved [10] to be extremely helpful in understanding the training-test split and which basic models I could implement.
Similar to scikit learn documentation, other websites and resources recommended linear regression-based models [4]. In addition to basic linear regression, they recommended using Lasso or Ridge regression to identify the effect of the regularization parameter.

3 Methodology

As discussed in the introduction, I hope to accurately predict future stock prices based on historical data. To this end, my primary exploration will be identifying the optimal features for price prediction. I will spend most of my time performing feature engineering. Then, I will use a few regression models to test the accuracy. I expect this to be an iterative process and, therefore, will perform three iterations to create my machine-learning model. Note that I will not describe the variables at this stage as I expect them to change over my three iterations. However, it is important to establish the predicted variable. It is defined as follows.

$$\mathrm{y} = \frac{\mathrm{close} - \mathrm{open}}{\mathrm{open}} \tag{1}$$

where y is labeled the daily change in price every day. It is normalized to reduce the trend of the data over time. The trend is described in detail later in the paper.
It is also important to note that predictors will be lagged by at least 1 day to ensure that only numbers that have already been computed are being used to calculate the daily change in price.
Furthermore, to test the model's usefulness, I will "invest" money using my predicted daily change and actual daily change from the test dataset to identify whether my model will return a profit or loss.

3.1 Time Period

There have been many fundamental shifts in Apple's strategy throughout its history; backtracking can result in unforeseen issues. At the same time, the data must be significantly large in size to ensure the different machine learning algorithms can be fitted correctly. Therefore, the time period I have decided to explore is from 2000 to 2024, offering approximately 6000 observations. During the initial dataset creation, I will backtrack a certain number of days to account for null values and will further describe this process in the feature engineering section.

3.2 Machine Learning Models

Evaluating the changes in each iteration requires that I have a standardized method of testing. I will implement the same three models in each iteration using their respective Python libraries. There were two primary considerations when deciding which machine learning models to use for testing: 1. I need to understand the interaction of the model with time-series data and ensure that I appropriately account for its unique effects. 2. I must be able to provide an overarching explanation of the math behind the algorithms. The models I will implement are linear regression, ridge regression, and lasso regression. I will explain further my understanding of these models in the Model Implementation section.

3.3 Training-Test split

The training dataset must be large enough to capture variability, and therefore, I plan on splitting the data 70:30 for training and testing, respectively. An important consideration when working with time-series data [5] is to ensure that the training data does not occur after the test data. This would lead to incorrect inferences as prices will be predicted depending on future prices. Therefore, random states will never be used, and data will be split manually.

4 Datasets

I will be using Yahoo's finance dataset, which can be imported using the "yfinance" library in Python. The dataset captures stock information during the working week daily and stores five features of interest per stock: (End, High, Low, Close, and Volume). I can extrapolate data for all the different stocks that I plan on considering through this Python library.

5 Feature Engineering

5.1 Seasonal Decomposition

To preface, a fundamental issue with time series data is that we need to identify a way to account for time not just as an index but as a variable. Changes in data are often correlated to the time at which they were recorded. As such, we must create variables that account for changes over time. These changes are broken down into three parts.
The first is called the "Trend" which reflects long-term increases or decreases in data [6]. The second is the "Seasonal" pattern which, as the name suggests, focuses on patterns that occur due to the season and can be broken down into different periods from weeks, months, to even years. The third pattern is the residual, which is any remaining randomness or noise that cannot be explained by the trend or seasonality.

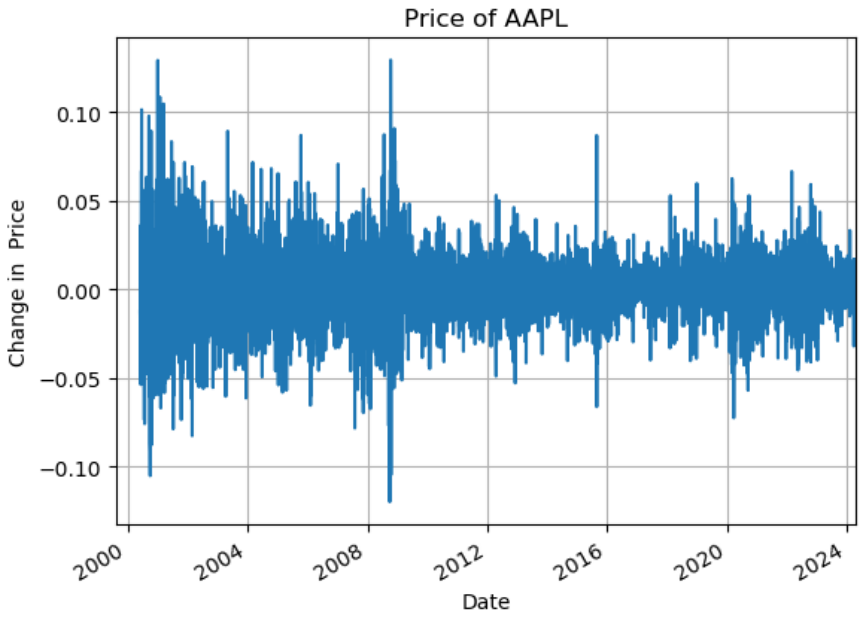

Figure 1: Apple Close Price vs Time.

Figure 1 suggests a limited trend, as ensured by normalization and seen through the graph. There is, however, seasonality, as indicated by the wavy movement of data throughout the dataset.
I began decomposition using a naive model, which provided additive and multiplicative methods.

$$Y[t] = T[t] + S[t] + e[t] \tag{2}$$

$$Y[t] = T[t] \cdot S[t] \cdot e[t] \tag{3}$$

Where T represents trend, S represents Seasonality, and e represents residuals [11]. The fundamental difference is that the multiplicative model aims to capture non-linear relationships. However, we cannot use a multiplicative model due to the negative values in the prediction variable. The seasonal decomposition works by first estimating the trend through a convolution filter, then removing the trend from the data and returning the average for each period in the de-trended data as the seasonal component. The results of the additive model can be seen below in Figure 2.

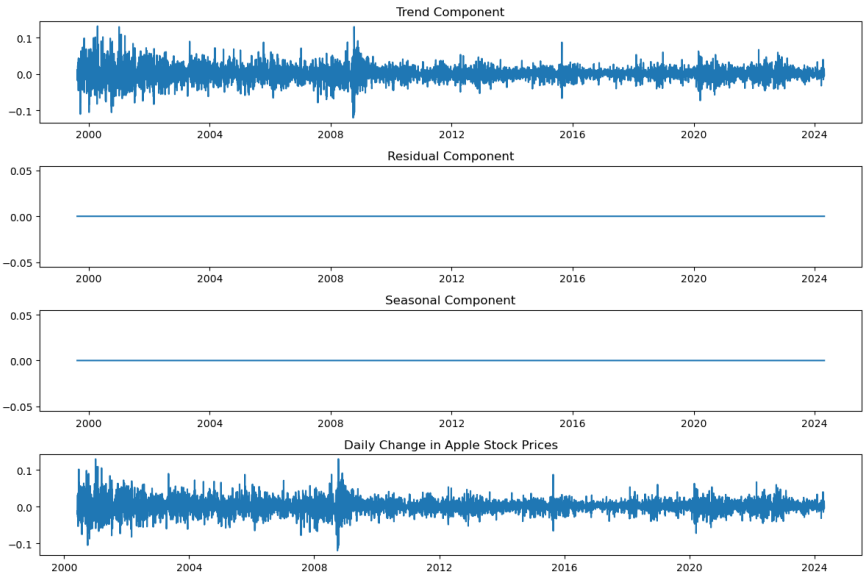

Figure 2: Naive Seasonal Decomposition

It is clear that the naivety of the model has failed to properly decompose the data into its appropriate sections and assumes that the entire dataset is composed of just the trend. As discussed in the statsmodels documentation [11], a more complex decomposition model needs to be considered, namely Season-Trend decomposition using LOESS. The results using LOESS were more promising and can be seen below in Figure 3.

Figure 3: LOESS Seasonal Decomposition

The seasonal and residual components have now been decomposed and better resemble what was expected. Looking at Figure 1, the seasonality resembles the variations we can observe, but due to normalization, the trend is not relevant to our model.

5.2 Feature Creation

Next, I created additional features to help improve the accuracy of my model. I will briefly discuss the meaning of each and what purpose they serve.
Rolling Averages are the average value of a certain time-period. They aim to "smoothen" short term fluctuations of data and focus on highlighting longer term trends. [12] I decided to taking rolling averages at periods of 50, 100, and 200 days.

5.2.1 Relative Strength Index

The relative strength index (RSI) is a technical indicator that measures the momentum of close prices. It is calculated using the following formula.

$$RSI = 1 - \frac{100}{1 + (\frac{g}{l})} \tag{4}$$

Where g is the average gain over n periods and l is the average loss over the same period.

5.2.2 Lags

A lag refers to using prices from a certain number of days prior to predict the price of the current day. To understand the optimal "lag" we can plot the autocorrelation function which calculates the correlation between a day n and all $n-i$ days where $i \in [0, n-1]$. For the close price of apple stocks the ACF has been plotted below.

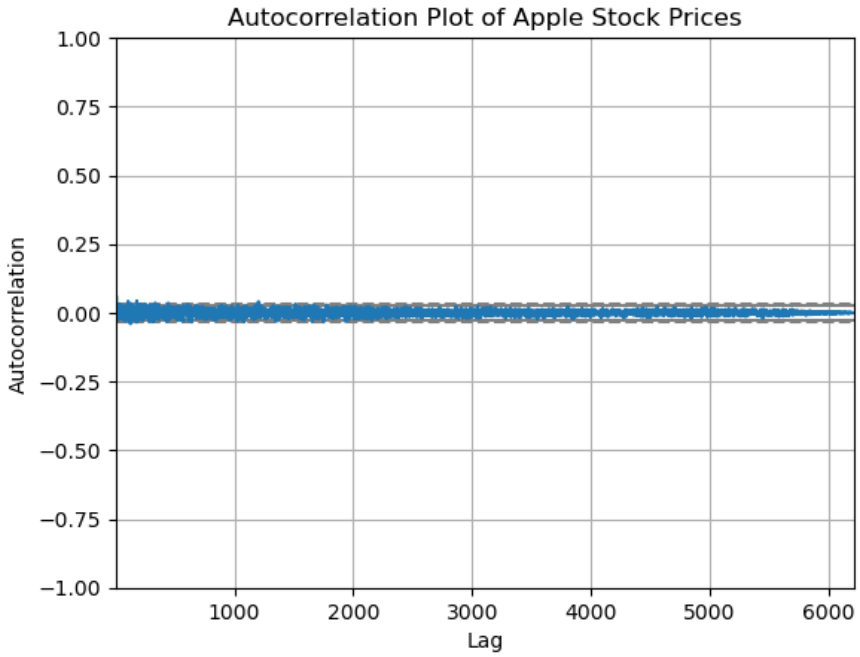

Figure 4: Autocorrelation Function for Close Prices

While no single lag value is highly correlated with the daily change, there is a slightly greater significance the closer we are to 0 days. Therefore my initial model will consider the lag of 1 and 5 days.

6 Model Implementation

The structure of this section will be as follows. I will begin with providing a brief overview of each of the three models I will implement. Then I will present my results for the first iteration. Based on my results, I will make changes to the features and once again test my models.

6.1 Linear Regression

The linear regression model can be understood as minimizing the following equation

$$\min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|_2^2 \tag{5}$$

y is a vector with the close prices of apple stock. X is a matrix of size n $\times$ f+1 where f is the number of x variables. θ is the coefficient of all the x variables and the constant intercept term. It is a column vector of size 1 $\times$ f+1. The goal of the linear regression model is to identify the parameters that minimize the distance between the target and predictor variables. It is the following equation:

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{6}$$

6.2 Ridge Regression

Ridge regression is very similar to linear regression but aims to regularize the data to reduce overfitting. This is to say, it creates a model that is able to better generalize to data that it has not seen before. We do this by minimizing on θ but adding a penalty term with respect to θ. The new minimization equation is as follows.

$$\min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|_2^2 + \lambda|\theta|_2^2 \tag{7}$$

All the variables denote the same quantities as those mentioned in the linear regression model and λ is the regularization parameter. It is a penalty term that biases θ to be closer to the origin. Minimizing with respect to θ provides the following equation.

$$\theta = (X^TX + \lambda I)^{-1}X^Ty \tag{8}$$

6.3 Lasso Regression

Lasso regression is very similar to ridge regression with the primary difference being that we utilize the L1-norm instead of the euclidean norm for the regularization parameter. We also penalize the absolute value instead of the square. Therefore the minimization function is:

$$\min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|_2^2 + \lambda|\theta|_1 \tag{9}$$

The optimization of the above function is made difficult due to the L1-norm and, therefore, must be done through coordinate descent. I believe that this is beyond the scope of this study and will therefore be skipping it.

6.4 Iteration 1

Data cleaning was not required for iteration 1 as its the cleaning that was done in the feature engineering section. Therefore I will be running the regression models and reporting the results in a table below. The main method of measuring the accuracy of my regression models is the r^2 score, Mean Squared Error(MSE) and returns. The r^2 score will be computed for the target data points predictions and their true values. The MSE will provide the average squared residuals for each data point. The returns will be calculated as described in the methodology section. The results of each model are presented in the table below.

Model	MSE	R^2 Score	Returns
Linear Regression	0.0001	0.4931	14.9196
Ridge Regression	0.0001	0.4167	14.4955
Lasso Regression	0.0002	-0.0023	1.6758

Table 1: Comparison of Models (Iteration 1)

6.5 Iteration 2

Table 1 showed that the linear and ridge regression models are good predictors with low mse and decent returns. However, the r^2 score is abnormally high for a quantitative model, indicating there is an error. The lasso regression model has the lowest accuracy with an r^2 score of -0.0023, indicating that the model was not well fit.
To identify which changes to make, I plotted the correlation matrix.

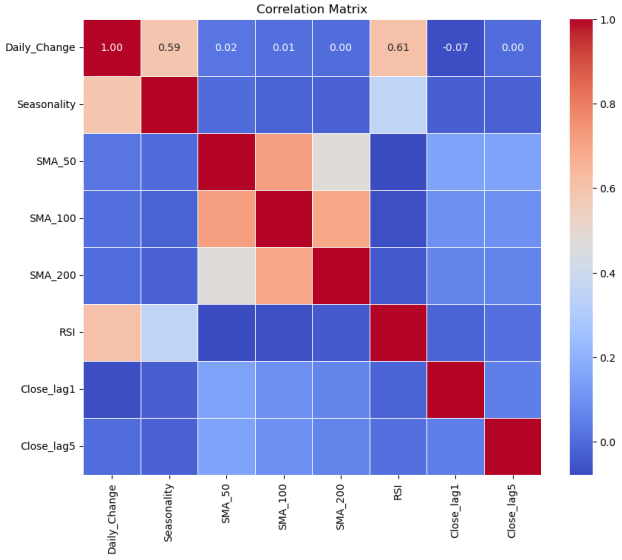

Figure 5: Correlation Matrix after iteration 1

As seen in figure 5 The RSI and Seasonality were very highly correlated relative to the rest of the features, suggesting there may be an error in its calculation. As such, I removed the seasonality and RSI as features. A potential issue of the seasonality was that it had not been lagged however upon taking its lag the accuracy dropped drastically to the point that it was just noise.
I added three additional features to help improve the accuracy of the model. They were the daily price changes of three prominent technology companies: Nvidia, Microsoft, and Amazon. The new model presented the following results.
As seen in Table 2, the new model has a far lower r^2 score and returns. This is likely a more accurate and

Model	MSE	R^2 Score	Returns
Linear Regression	0.0002	0.0019	1.6764
Ridge Regression	0.0002	0.0024	1.5199
Lasso Regression	0.0002	-0.0023	1.6758

Table 2: Comparison of Models (Iteration 2)

each feature has a far lower correlation with the target and with each other further indicating less error.

6.6 Iteration 3

In iteration 3, the primary objective was to identify marginal improvements to the models to help fine-tune and increase the r^2 score. Therefore, more lags were added, and the rolling average was removed. The results can be seen in the below table.

Model	**MSE**	**R^2** Score	**Returns**
Linear Regression	0.0002	0.0011	1.8706
Ridge Regression	0.0002	0.0025	2.2729
Lasso Regression	0.0002	-0.0023	1.6764

Table 3: Comparison of Models (Iteration 2)

The differences in iterations 2 and 3 are marginal. However, the ridge regression model has performed better in terms of returns and r^2 score.

6.7 Results

Across the three iterations, it is clear that the returns using strictly technical indicators are minimal. The highest returns we could achieve over a roughly 1800-day period was 2.2729. Essentially, for every 1$ investment we would see 2.2729$ returns. While this may seem extremely high, we need to subtract the average return of index funds to account for market changes. This is a limitation of the study and the accounting for market changes is beyond the scope of the paper.

This talks to the difficulty associated with experiencing meaningful results using simple ML models in quantitative trading. Most solutions with a high r^2 score or returns likely had errors as many companies focus tremendous manpower to create these models, and, therefore, it would be unlikely that I discover a more successful algorithm.

In the context of the study, there were a few interesting results. Lasso regression was the least susceptible to change, displaying similar returns, MSE, and r^2 scores in all three iterations. Ridge and Linear regression were more susceptible to change but produced near identical results. The small difference could be associated with the regularization parameter λ playing a role in providing more accurate results.

7 Conclusion

Overall, the study allowed me to better understand quantitative topics in the context of a real problem. It also helped me better understand regression models and how to set up a supervised machine-learning model. I also had an opportunity to gauge the unique challenges to time-series data. Namely performing seasonal decomposition, splitting data to ensure training data occurs before testing data, and feature engineering. In terms of feature engineering, I have come to realize the importance of normalizing data. Without normalization, companies such as Apple would consistently suggest returns are increasing due to the general trend associated with it; however, on a larger scale, this would create an unsustainable model as not all companies operate in such a manner. I also understand that a higher correlation between features may not necessarily suggest a better model but perhaps an error. In this study, I aimed for features with a correlation between -0.2 and 0.2 before considering an error in my setup. I now have a better understanding of supervised machine learning in the context of time-series and quantitative finance.

Github Repository

References

[1] Aniruddha Bhandari. *Multicollinearity — Causes, Effects and Detection Using VIF*. 2024. URL: https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/.

[2] Vitor Cerqueira. *3 Types of Seasonality and How to Detect Them*. 2024. URL: https://towardsdatascience.com/3-types-of-seasonality-and-how-to-detect-them-4e03f548d167.

[3] Vitor Cerqueira. *8 Techniques to Model Seasonality*. 2024. URL: https://towardsdatascience.com/8-techniques-to-model-seasonality-2f81d739710.

[4] Michael Foley. *Chapter 3 Time Series Regression*. 2021. URL: https://bookdown.org/mpfoley1973/time-series/regression.html.

[5] GeeksforGeeks. *Time Series Analysis Visualization in Python*. 2024. URL: https://www.geeksforgeeks.org/time-series-data-visualization-in-python/?ref=lbp.

[6] Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2018.

[7] Neha Agarwal Kunal Pahwa. "Stock Market Analysis using Supervised Machine Learning". In: *IEEE* (2019).

[8] Marvin Lanhenke. *Understanding the Covariance Matrix*. 2024. URL: https://towardsdatascience.com/understanding-the-covariance-matrix-92076554ea44.

[9] Robert Nau. *Stationarity and differencing*. 2024. URL: https://people.duke.edu/~rnau/411diff.htm.

[10] Scikit-Learn. *Time-related feature engineering¶*. 2021. URL: https://scikit-learn.org/stable/auto_examples/applications/plot_cyclical_feature_engineering.html.

[11] statsmodels. *seasonal$_d$ecompose*. 2024. URL: https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html.

[12] Indeed Editorial Team. *Rolling Averages: What They Are and How To Calculate Them*. 2022. URL: https://www.indeed.com/career-advice/career-development/what-is-rolling-average.

[13] Rufai Yusuf Zakari Zaharaddeen Karami Lawal Hayati Yassin. "Stock Market Prediction using Supervised Machine Learning Techniques: An Overview". In: *IEEE* (2020).