Forecasting Prices Using Stock Market Index Data

Student Number: 690065435

December 2022

1 Introduction

Stock markets from across the world are tracked using indices that measure a section of the stock market, such as the Nasdaq Composite Index, which is a market capitalisation-weighted index [1]. Price forecasting is an important task in the financial industry as it can be used to inform strategies.

Historically, financial institutions have used discretionary methods to make investment decisions – they rely on fundamentals and the judgement of analysts [2]. However, with the rise of big data and computational power, systematic methods have become increasingly popular – institutions use rules-based strategies that are implemented by a computer and involve little human intervention [2]. Systematic methods enable decisions to be made quickly, which leading market makers and high-frequency trading firms such as Jane Street Capital and Hudson River Trading use to exploit arbitrage opportunities and maximise profits by trading at high volumes [3]. However, these firms do not publicly disclose their strategies, which makes it hard to understand how they make decisions.

In this project, I propose the following question: can we use regression models on stock market index data to forecast prices effectively? My initial hypothesis is that this is not possible as the stock market is a complex system that is difficult to predict, but we will be able to predict general trends. I will explore the stock exchange dataset, describe our methodology, train three regression models, and discuss the outcomes.

Previous studies have investigated the prediction of stock market trends using regression on moving averages [4], but they focused on individual stocks whereas I am focusing on a stock market index. This is a potential snag, as the index is not a direct representation of the stocks it tracks. The efficient market hypothesis also suggests a potential snag, as it states that asset prices reflect all available information, yet we are only predicting on a subset of information [5].

2 Methodology and Dataset

2.1 About Stock Exchange Dataset

We are using a stock exchange dataset on Kaggle that was collected from Yahoo Finance and contains the daily price data for stock market indices across the world. Each record in the CSV file contains the following information for each trading day:

- Index
- Date
- Opening price
- Highest price
- Lowest price
- Closing price adjusted for splits
- Adjusted closing price adjusted for both dividends and splits
- Volume traded

2.2 Data Cleaning

I loaded the CSV file into a pandas dataframe and converted the date to a datetime64 object to make it easier to work with. It originally consisted of 112457 rows. After looking at the information of the data, I noticed that there were 2204 incomplete rows, so I removed them to ensure that the data was consistent. There were also rows with trading volume recorded as 0 in earlier dates where this information was not available, so I removed them to prevent them from skewing the distribution of volume data – this further reduced the dataset from 110253 rows to 68160 rows.

2.3 Data Exploration

The dataset contains a separate CSV file that describes the index information. I loaded this into a pandas dataframe to see the stock exchange that each index corresponds to. Then, I split the data by index, as each index has different characteristics and may behave differently, and counted the sample size of each index. I decided to focus on the Nasdaq Composite Index (IXIC) as it has the most data points.

```
000001.SS: 4430
399001.SZ: 4190
GDAXI: 5464
GSPTSE: 9183
HSI: 4890
IXIC: 9233
KS11: 6029
N100: 3817
N225: 4635
NSEI: 2026
NYA: 5115
SSMI: 4632
TWII: 4516
```

Figure 1: Sample Size Per Index in Dataset

Next, I calculated the distribution of features to understand patterns in the data. The histograms show the distributions of open, low, high, and adjusted close are extremely similar – this makes sense as open, close, and adjusted close are between the low and high for each day.

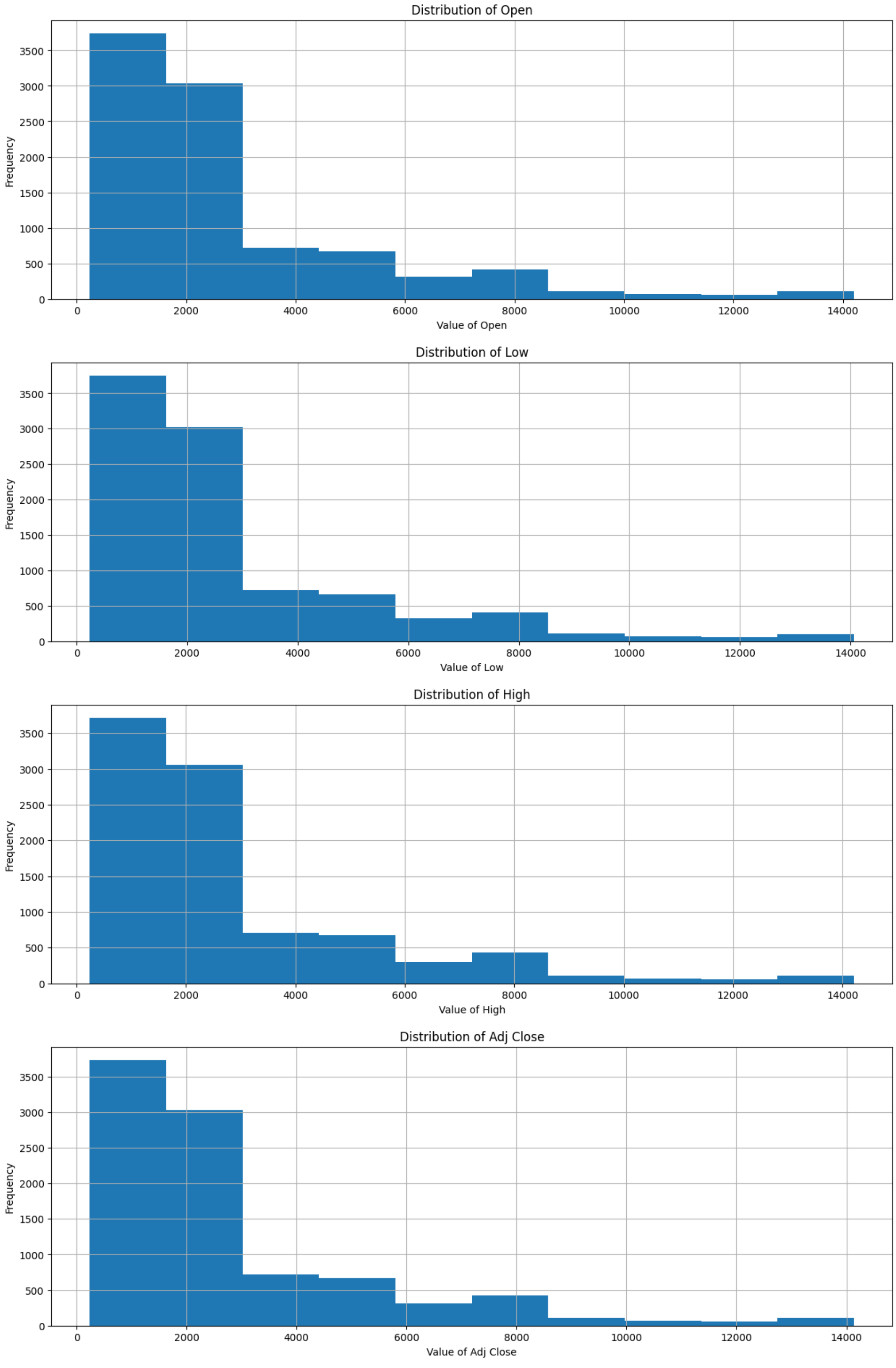

Figure 2: Distribution of Open, Low, High, and Adjusted Close in Dataset

Meanwhile, the graph of daily volume traded over time shows that market activity has increased over time, which can be attributed to the rise of electronic trading making the stock market more accessible to everyone, including high-frequency trading firms.

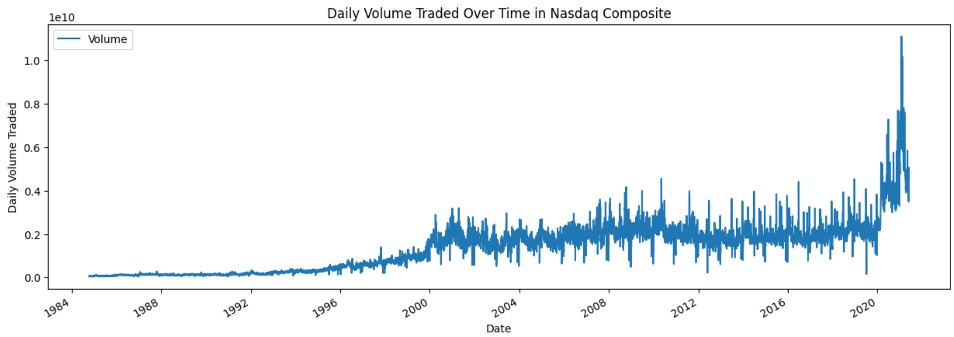

Figure 3: Daily Volume Traded Over Time in Nasdaq Composite

I also used Pearson's correlation coefficient to find the correlation between features and the adjusted close price, which is our target variable. As expected, the open, low, and high prices are highly correlated with the adjusted close price because they represent the variance of the index's price over the day. Volume appears to be correlated to adjusted close price, but it coincides with the rise of electronic trading, so it is not a useful feature for our model.

Low	0.999934
High	0.999924
Open	0.999872
Volume	0.784782

Figure 4: Pearson's Correlation Coefficients of Features against Adjusted Close in Nasdaq Composite

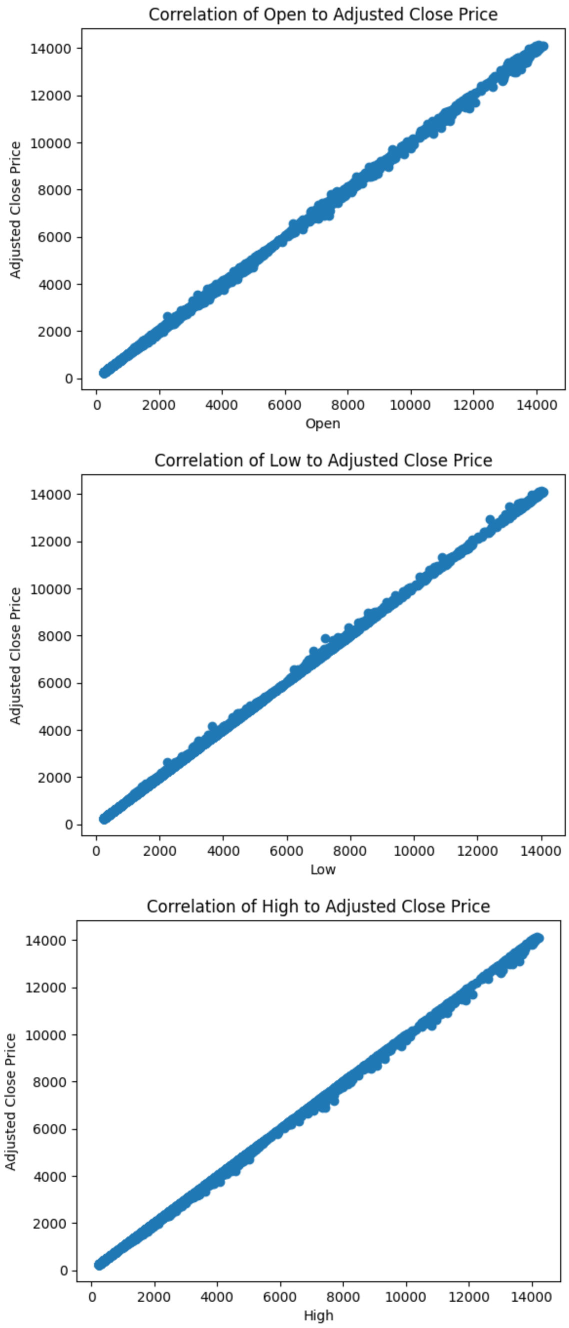

Figure 5: Correlation of Open, Low, and High against Adjusted Close in Nasdaq Composite

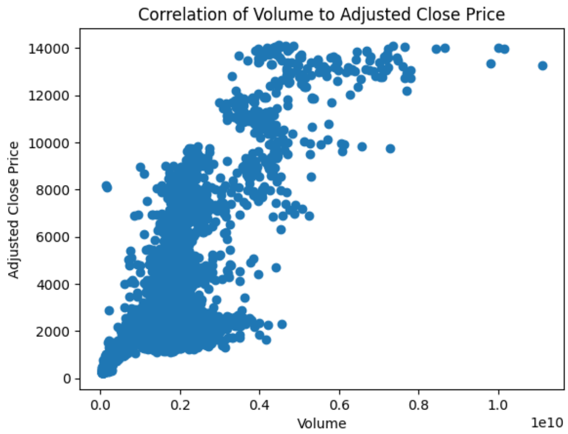

Figure 6: Correlation of Volume against Adjusted Close in Nasdaq Composite

2.4 Feature Engineering

The moving average is a data smoothing technique that is used to track price movements by plotting the average prices over a period of time [6]. It helps to remove noise from random short-term price fluctuations in data to highlight the overall trend direction [6].

Simple moving averages give equal weight to each point, whereas exponential moving averages give more weight to recent points. Choosing the time period is a trade-off between noise and accuracy – a longer period will smooth the data more at the expense of accuracy. I applied these techniques on each index to show how it works with different price patterns.

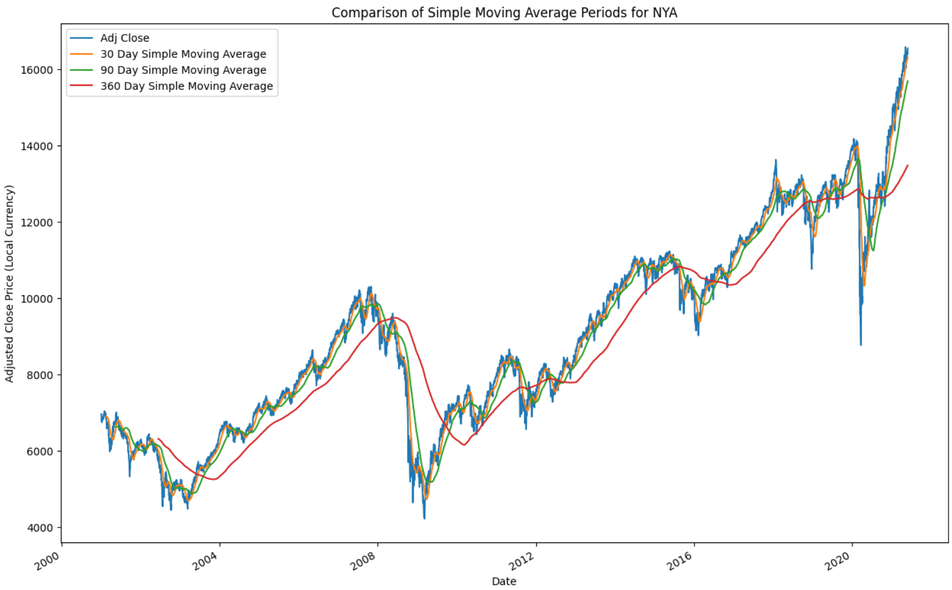

Figure 7: Comparison of Simple Moving Averages for NYA

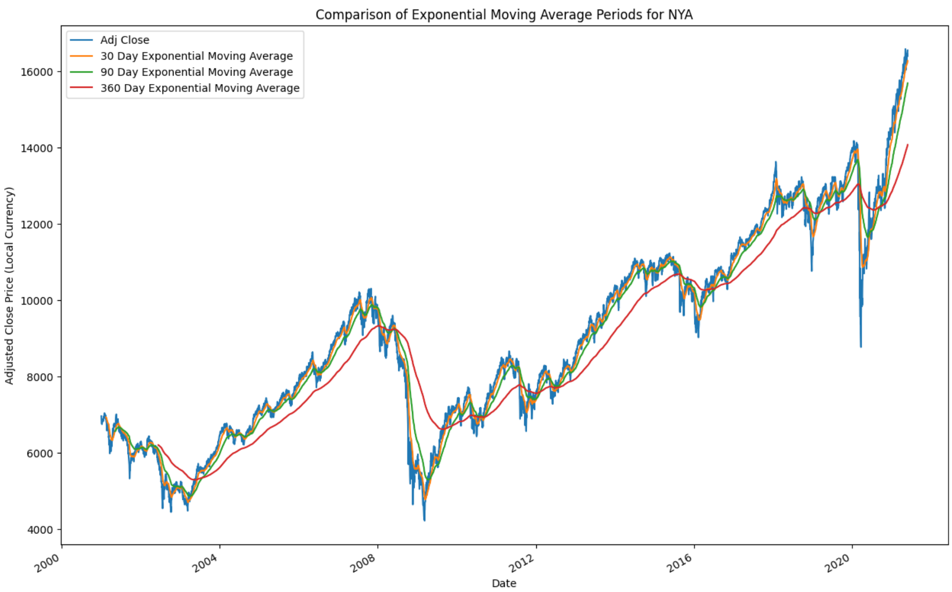

Figure 8: Comparison of Exponential Moving Averages for NYA

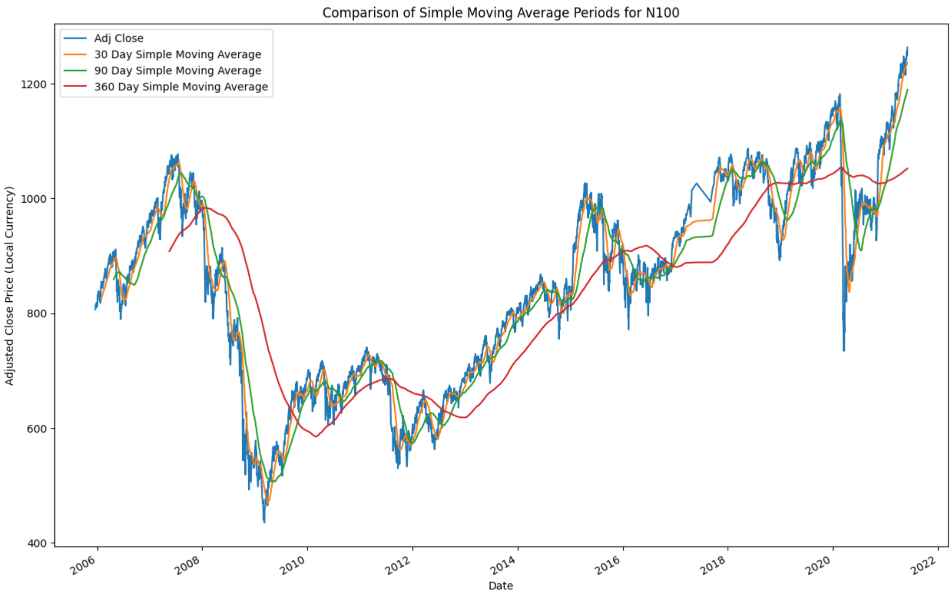

Figure 9: Comparison of Simple Moving Averages for N100

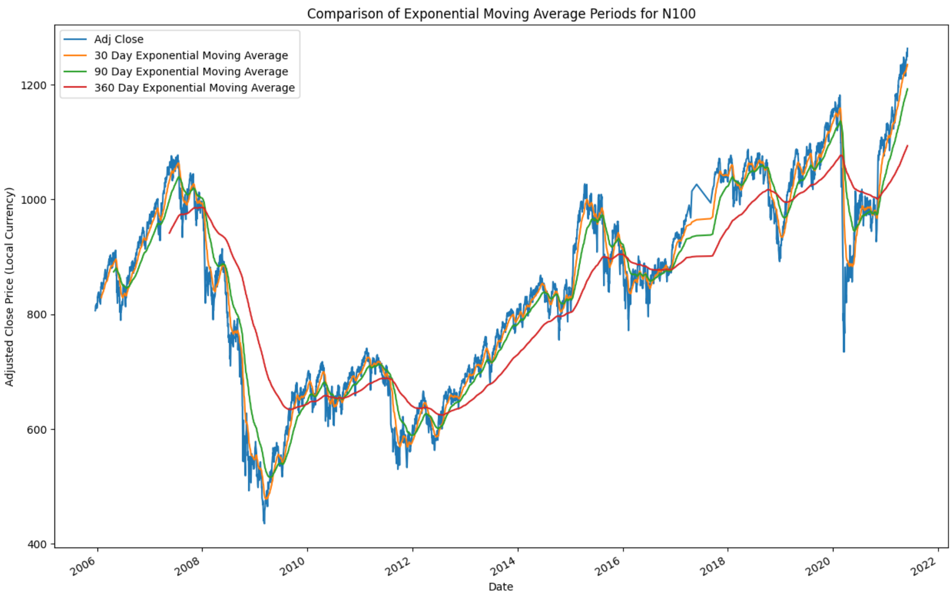

Figure 10: Comparison of Exponential Moving Averages for N100

2.5 Data Filtering

I compared the simple and exponential moving averages over 30, 90, and 360 day periods on the IXIC data. The 360 day simple and exponential moving averages were poor indicators of the adjusted close price, as they lost too much information and tracked price movements poorly. Thus, I removed them from the dataset and focused on the 30 day and 90 day simple and exponential moving averages.

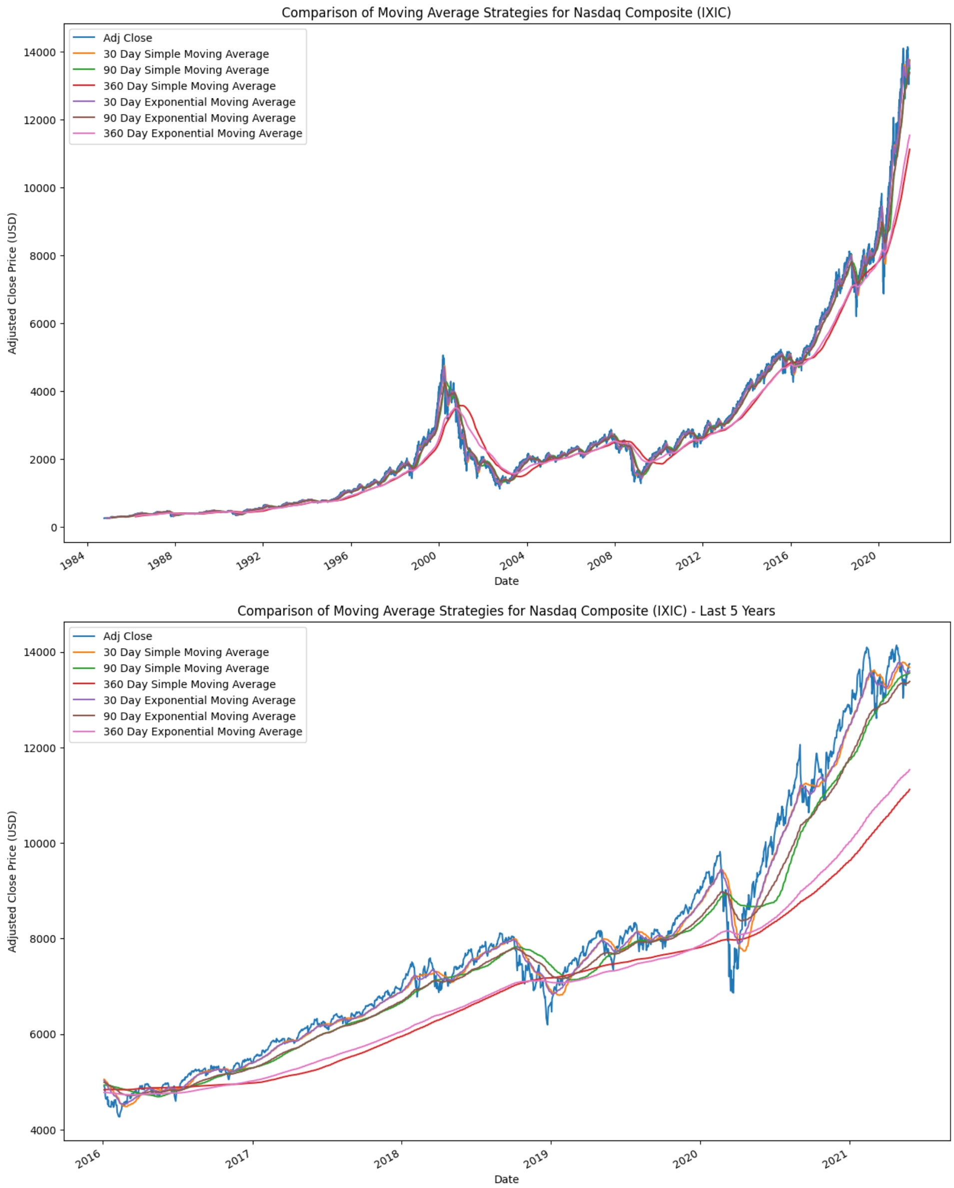

Figure 11: Comparison of Moving Average Strategies for Nasdaq Composite

2.6 Data Preprocessing

I decided to focus on predicting the adjusted close price 30 days in advance, as this would be useful to investors who want to invest part of their monthly paycheck. Thus, I shifted the values of the four input variables to the future by 30 days to remove look-ahead bias [7].

Then, I split the data into training and testing sets, with 80% of the data being used for training and 20% for testing the performance of our models.

2.7 Regression Models

I will apply three different regression models to compare their price prediction performance.

Ridge regression is a linear regression model that uses L2 regularisation to reduce the variance of the model. It is useful for reducing overfitting and improving the generalisation of the model.

LASSO regression is another regularised linear regression model that uses L1 regularisation to reduce the variance of the model. It is useful for feature selection, as it can remove features that are not important to the model.

Polynomial regression is a linear regression model that uses polynomial features to fit a non-linear relationship between the input and target variables.

2.8 Model Evaluation

For each regression model, I will calculate the R^2 score, which is the proportion of the variance in the target variable that is predictable from the input variable. The closer the R^2 score is to 1.0, the better the model is at predicting the target variable. We will compare this score to the R^2 score of the naive benchmark that predicts adjusted close price based on the last price of the index.

3 Results

3.1 Ridge Regression

I performed hyperparameter tuning on the ridge regression model using grid search with five-fold cross validation on several values of alpha between 0.1 and 1000.0. I found the optimal alpha value was 0.1, achieving a mean test score of 0.652964, although the mean test score varied very little between alpha values.

Alpha	Mean Test Score
0.1	0.652964
0.2	0.652964
0.3	0.652964
0.4	0.652964
0.5	0.652964
0.6	0.652964
0.7	0.652964
0.8	0.652964
0.9	0.652964
1.0	0.652964
5.0	0.652964
10.0	0.652964
25.0	0.652963
50.0	0.652962
75.0	0.652961
100.0	0.652961
500.0	0.652947
1000.0	0.652931

Figure 12: Hyperparameter Tuning for Ridge Regression

With an alpha value of 0.1, ridge regression achieved an R^2 score of 0.95623 when predicting on the test set. The minimal difference in mean test scores between alpha values suggests that a linear regression model would give a similar R^2 score.

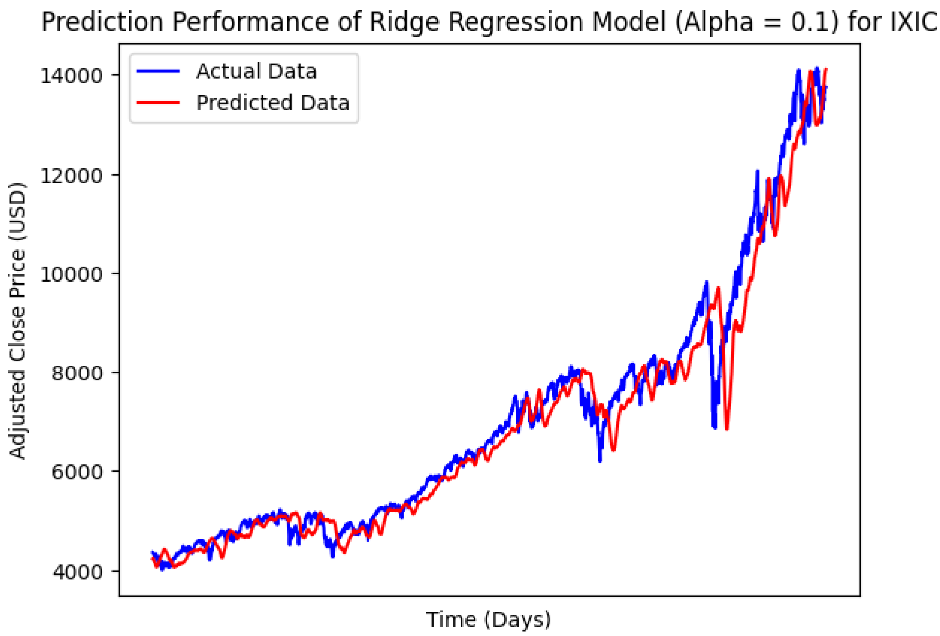

Figure 13: Prediction Performance of Ridge Regression Model

3.2 LASSO Regression

I performed hyperparameter tuning for the LASSO regression model in the same way as ridge regression, with alpha values ranging from 0.1 to 1000.0. However, the mean test scores varied a lot more, and I found the optimal alpha value to be 1000.0, achieving a mean test score of 0.616147.

Alpha	Mean Test Score
0.1	0.566200
0.2	0.566211
0.3	0.566222
0.4	0.566233
0.5	0.566245
0.6	0.566256
0.7	0.566267
0.8	0.566278
0.9	0.566289
1.0	0.566301
5.0	0.566749
10.0	0.567308
25.0	0.568976
50.0	0.571728
75.0	0.574429
100.0	0.577083
500.0	0.597556
1000.0	0.616147

Figure 14: Hyperparameter Tuning for LASSO Regression

With an alpha value of 1000.0, LASSO regression achieved an R^2 score of 0.95243 when predicting on the test set, which was slightly worse than ridge regression. Both ridge and LASSO are regularised linear regression models, so it is not surprising that they performed similarly. In the graph we can see that it was less sensitive to changes in the direction of the price, so it fared better in stable periods but worse in volatile markets.

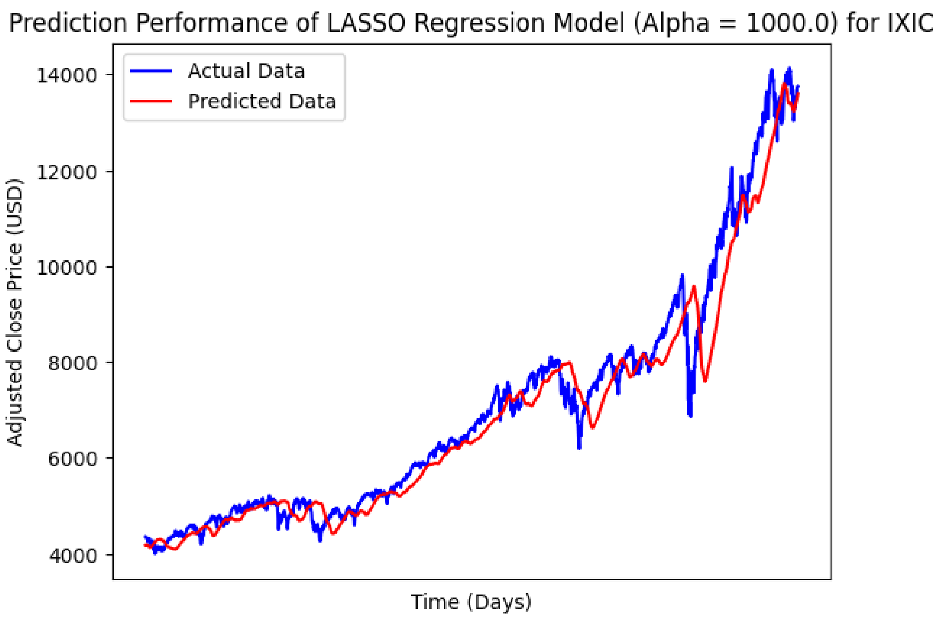

Figure 15: Prediction Performance of LASSO Regression Model

3.3 Polynomial Regression

For polynomial regression, I used a preprocessing pipeline to generate a feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the specified degree before fitting the model. I performed hyperparameter tuning on the degree of the polynomial features, testing each degree between 2 and 8 – the optimal degree was 2, achieving a mean test score of 0.804763.

Degree	Mean Test Score
2	8.047632e-01
3	3.682671e-01
4	-4.126681e+01
5	-2.445082e+03
6	-2.542428e+04
7	-5.283752e+05
8	-1.262192e+07

Figure 16: Hyperparameter Tuning for Polynomial Regression

Polynomial regression performed poorly, achieving a R^2 score of only 0.57730 on the test set. This was significantly worse than the other two models, which suggests that there is a linear relationship between the input and target variables and the model was overfitting.

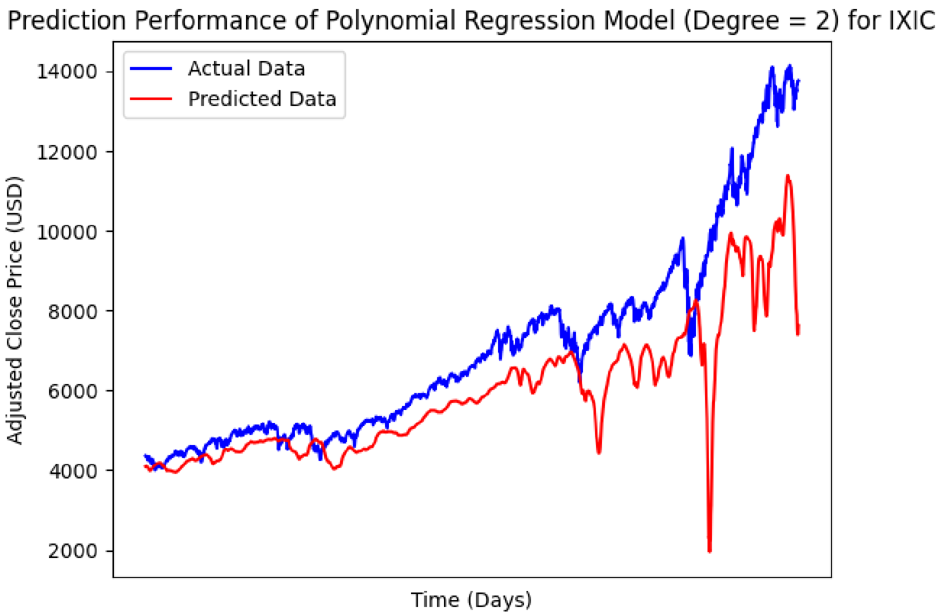

Figure 17: Prediction Performance of Polynomial Regression Model

4 Discussion

4.1 Comparison to Last Value Naive Benchmark

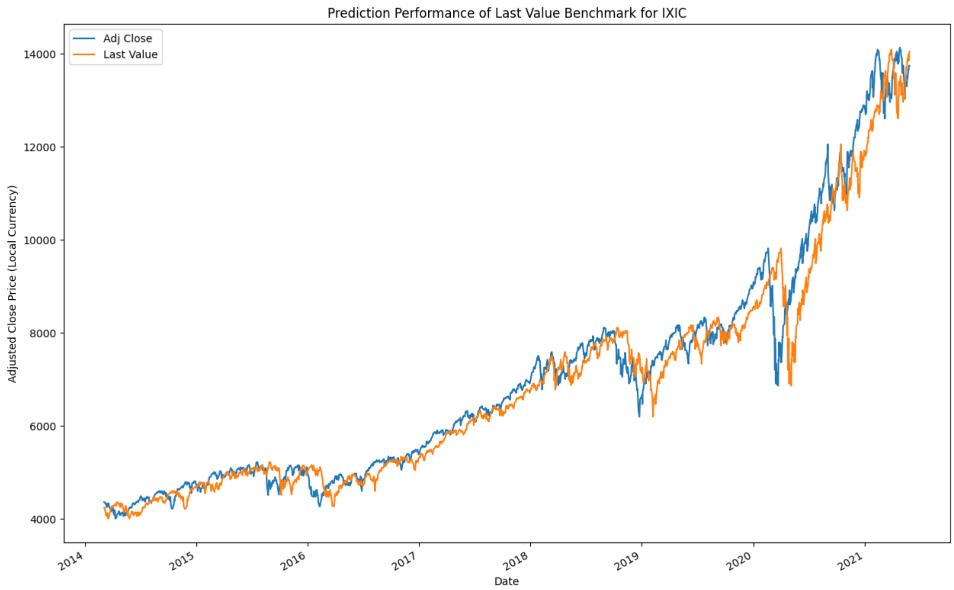

Figure 18: Prediction Performance of Last Value Benchmark

To evaluate the usefulness of our results, we implemented the last value benchmark, which predicts the adjusted close price 30 days ahead by using the last value of the adjusted close price. It performed slightly better than our models, achieving an R^2 score of 0.96029. This suggests that our models are ineffective at forecasting prices on the Nasdaq Composite Index, and that using moving averages resulted in too much information loss. However, it may still be useful to predict general trends – whereas the last value benchmark has a consistent lagging sentiment and is prone to outliers, our models can be more reactive to changing sentiment and less prone to outliers.

Model	R^2 Score
Ridge Regression	0.956233
LASSO Regression	0.952436
Polynomial Regression	0.577303
Last Value Benchmark	0.960294

Figure 19: Comparison of Prediction Performance of Models

4.2 Limitations of the Study and Future Work

The efficient market hypothesis states that asset prices reflect all available information [5], which makes it difficult to get arbitrage opportunities. However, Renaissance Technologies, a hedge fund that uses systematic methods, serves as a perfect counter-example – their Medallion Fund has achieved 66.07% annualised returns since 1988 [8]. This suggests that it may be possible to achieve better results with higher quality data. Examples of this include data aggregated on a minute by minute basis rather than daily, and counting the number of buy and sell orders on the market at a given moment.

Furthermore, the stock market involves game theory between market participants [9] – prices are an equilibrium between people going long and short [10], so any signals that provide good predictions for future prices will be quickly discovered and priced in [11]. This means that models that are successful on backtested data are unlikely to be successful in the future.

Future analysis could focus on logarithmic returns as the target variable – this may be a more relevant measure for investors, as it represents relative changes in prices rather than absolute changes. It is also symmetric and additive over time [12], which could make it easier to develop a well-performing regression model.

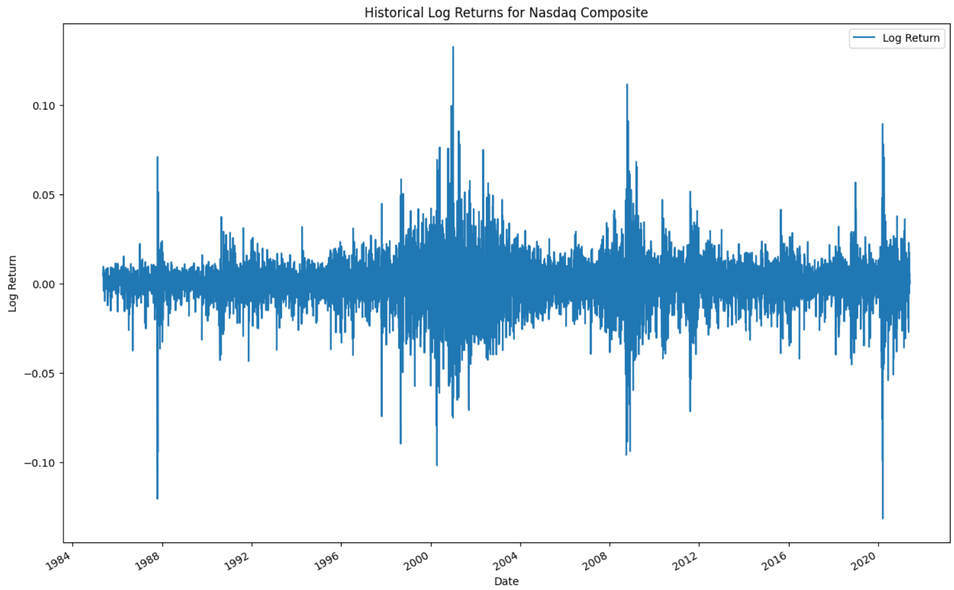

Figure 20: Historical Log Returns for Nasdaq Composite

References

[1] Nasdaq, "Nasdaq Composite Index." https://indexes.nasdaqomx.com/docs/methodology_COMP.pdf, 2020. Date Accessed: 20 December 2022.

[2] C. R. Harvey, S. Rattray, A. Sinclair, and O. Van Hemert, Man vs. machine: Comparing discretionary and systematic hedge fund performance *The Journal of Portfolio Management*, vol. 43, no. 4, pp. 55–69, 2017.

[3] I. Aldridge, *High-frequency trading: a practical guide to algorithmic strategies and trading systems*, vol. 604. John Wiley & Sons, 2013.

[4] S. Dinesh, N. Rao, S. Anusha, and R. Samhitha, Prediction of Trends in Stock Market using Moving Averages and Machine Learning in *2021 6th International Conference for Convergence in Technology (I2CT)*, pp. 1–5, IEEE, 2021.

[5] E. F. Fama, Efficient capital markets: A review of theory and empirical work *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[6] C. Mitchell, "Moving Average Chart." https://www.investopedia.com/terms/m/movingaverage.asp, 2022. Date Accessed: 20 December 2022.

[7] G. Baquero, J. Ter Horst, and M. Verbeek, Survival, look-ahead bias, and persistence in hedge fund performance *Journal of Financial and Quantitative analysis*, vol. 40, no. 3, pp. 493–517, 2005.

[8] B. Cornell, Medallion Fund: The Ultimate Counterexample? *The Journal of Portfolio Management*, vol. 46, no. 4, pp. 156–159, 2020.

[9] F. Allen and S. Morris, Finance applications of game theory 1998.

[10] A. V. Thakor, Game theory in finance *Financial Management*, pp. 71–94, 1991.

[11] D. Carfì, F. Musolino, *et al.*, Fair redistribution in financial markets: a game theory complete analysis *Journal of Advanced Studies in Finance*, vol. 2, no. 2, p. 4, 2011.

[12] L. Loura, "Why Use Logarithmic Returns In Time Series Modelling." https://lucaslouca.com/Why-Use-Logarithmic-Returns-In-Time-Series-Modelling/, 2021. Date Accessed: 20 December 2022.