

Analysis and Forecasting the Price of The S&P 500 Index Using the Arima Model

[http://doi.org/10.61093/fmir.7\(4\).113-134.2023](http://doi.org/10.61093/fmir.7(4).113-134.2023)

Vadym Dun,

Sumy State University, student, Sumy, Ukraine

Research supervisor: Serhiy Lyeonov, Dr., Prof Department of Economic Cybernetics, Sumy State University, Ukraine

Abstract. *The results of the research allowed to determine that the chosen S&P 500 index can serve as a reflection of the state and forecasts of economic development of the United States. A successful forecast of the index can serve not only as a key point in building an individual investment strategy, but also as an indicator of the general state of the economy. The mathematical model for predicting the dynamics of the index was built. Through exploratory data analysis, a better understanding of the time series and its characteristics was obtained. The application of various statistical methods, such as moving statistics and stationarity tests, made it possible to identify trends and seasonality in the data. Seasonal decomposition and logarithmic transformation helped to better understand the contribution of each component to the overall index dynamics, and special attention was paid to the stationary ADF test, where was considered not only the code but also the significant formulas. The optimal selection of parameters was done automatically. ARIMA model showed good results - the evaluation of the model accuracy included the comparison of the predicted values with the actual values of the SP500 index, both visually and using several metrics - MAE, MSE, RMSE, MAPE. The result of the work is a model for predicting the dynamics of the S & P 500 index, implemented using the Python programming language with a MAPE of about 1.9%, the accuracy of the model is 98.1%. and such good results indicate the possibility of using this tool by market participants in real conditions.*

Keywords: time series forecasting, stock market, investments, Python, S&P 500 index, stationarity test, ARIMA model, moving statistics.

JEL Classification: G00, G12, G17, G19.

Received: 8.10.2023

Accepted: 14.12.2023

Published: 31.12.2023

Funding: There is no funding for this research.

Publisher: Academic Research and Publishing UG (i. G.) (Germany)

Cite as: Dun, V. (2023). Analysis and Forecasting the Price of the S&P 500 Index Using the Arima Model. *Financial Markets, Institutions and Risks*, 7(4), 113-134. [http://doi.org/10.61093/fmir.7\(4\).113-134.2023](http://doi.org/10.61093/fmir.7(4).113-134.2023)



Copyright: © 2023 by the author. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Introduction

One of the main goals of econometric modeling of the money market is the study of time series in finance. For a long time, financial market researchers assumed that financial assets follow a normal distribution and are completely unpredictable. However, the application of new approaches to financial market modeling has shown that real time series of financial data are not only devoid of randomness, but also have a long memory. This means that past events have a strong influence on the future returns of financial assets.

Stock market indices are an integral part of the analysis and understanding of the financial system. They provide macroeconomists and financial economists with important tools for studying and forecasting market behavior and economic development. Without reliable and consistent indices, it becomes more difficult to identify long-term patterns, evaluate the performance of financial companies, and make comparisons between different markets.

Financial indices are an important source of information for traders and investors. They provide a quick

summary of the state of stock markets, evaluate their performance and make informed investment decisions. Indices allow traders and investors to easily track changes in the market, identify trends and predict possible price movements. However, despite the importance of indices, not enough attention is always paid to their methodology and proper use. Currently, information on index methodology is rarely included in economics or business curricula and remains available only to a limited number of specialists. This can lead to misinterpretation of data and potentially misleading index-based decisions (Kupper, 2022).

Historically, there have been many different indexes, offering different calculation methodologies and focusing on different aspects of the market. Some have become widely known and used, such as the Dow Jones Industrial Average (DJIA), the S&P 500, and the NASDAQ Composite [14]. However, each index has its own characteristics and purpose, and the choice of a particular index depends on your analysis or investment strategy. Given these factors, it is clear that a thorough understanding of indices and their methodology is key to the proper use and interpretation of data. The correct use of indices allows you to draw more accurate and meaningful conclusions about the state of the stock markets and to make more informed decisions.

According to the calculation method, the indices are divided into groups, the most common of which are as follows (Investopedia, 2023):

- Price indices: Calculated by averaging the prices of index components with their weights. Examples include the Dow Jones Industrial Average (DJIA) and the Nikkei 225.
- Market-cap-weighted indices: Calculated by taking into account the market value of the index components. The larger the market capitalization of a company, the greater its weight in the index. Examples include the S&P 500 and the NASDAQ Composite.
- Balanced indices: All index constituents are equally weighted, regardless of size or market capitalization.
- Factor indices: These are calculated based on specific factors such as value factor (price/earnings), capitalization factor (small, mid, or large capitalization), asset value factor, and others. Examples of factor indices include the Fama-French three-factor model and the MSCI Minimum Volatility Index.

In the stock market, indices provide a number of important functions, including:

- The first function of indices in the stock market is their ability to reflect the overall performance of the market. By combining several stocks or other financial instruments into a single index, indices allow investors to measure overall market movements and changes. Indices such as the S&P 500 or the Dow Jones Industrial Average provide valuable information about the market and its long-term trend (Ellis, 2016).
- The second function of indices in the stock market is to be used as a benchmark to compare the performance of investment portfolios, mutual funds or individual stocks. Indices allow investors to gauge how well their investments are performing relative to the overall market. Comparison to an index can help determine the effectiveness of an investment strategy and whether it needs to be adjusted (Ganeshwaran, 2022).
- The third function of indices in the stock market has to do with their ability to serve as a market trend indicator. Changes in the index can indicate current market trends. For example, an increase in an index can be a signal of a strong economy and investor confidence, while a decrease in an index can indicate economic problems or uncertainty in the market. Such signals can help investors make decisions to buy or sell assets based on current market trends (Novotný, Jaklová, 2022).
- The fourth function of indices in the stock market is their ability to guide investment strategies. Indices provide information about industries, geographic regions, or other market segments. This information can be used by investors to develop their investment strategies and make asset allocation decisions. For example, an investor may decide to focus on an industry sector that is performing well relative to the overall market (Ellis, 2016). One of the most interesting assets is the S&P 500 index, which is one of the main indicators of the American economy. Successful and accurate forecasting of the index allows analysts and economists to draw conclusions about the trends in the economy and take appropriate actions not only within an individual investment portfolio, but also within countries.

The S&P 500 index is designed to measure the performance of eligible stocks listed on the NYSE and Nasdaq. It is weighted by float-adjusted market capitalization and incorporates liquidity and tradability criteria in the constituent selection process (S&P Global, 2023). The S&P 500 Index measures the value of the stocks of the 500 largest companies by market capitalization listed on the New York Stock Exchange or Nasdaq. The intent of Standard & Poor's is to have a price that provides a quick look at the stock market and the economy (S&P Global, 2023).

In order to be selected by the Index Committee and to be included in the S&P 500 Index, a company must meet certain criteria (Corporate Finance Institute, 2023):

- Geographic location: The company must be incorporated and headquartered in the United States.
- Market Capitalization: The company must have a market capitalization of at least \$8.2 billion. Market capitalization is calculated by multiplying the company's current share price by the total number of shares outstanding.
- Stock Liquidity: A company's stock must be highly liquid, meaning that it is actively traded on the stock exchange. This makes it easy to buy and sell a company's stock in the market.
- Public availability: At least 50 percent of a company's outstanding shares must be publicly traded. This makes the company's stock widely available in the marketplace and allows investors to include it in their portfolios.
- Financial performance: The company must have positive earnings in the most recent quarter and positive earnings in the previous four quarters. This indicates that the company is financially stable and successful.

It is important to note that while the S&P 500 Index was originally intended to be an index of 500 companies, it actually contains 505 stocks. This is because some companies, such as Google (now Alphabet Inc.), Facebook, and Berkshire Hathaway, have multiple share classes that are still considered separate components of the index (S&P Global, 2023).

The S&P 500 Index is an important part of the stock market and its connection to the U.S. economy is obvious. This index is considered by most analysts and investors as an indicator of the overall health of the stock market. It includes the 500 largest publicly traded U.S. companies representing various industries and sectors of the economy. Therefore, changes in the index effect not only the performance of individual companies, but also the collective performance of the economy as a whole. Also an important tool for passive investors seeking access to the states economy through index funds. Index funds, such as ETFs or index mutual funds, track the performance of the index and allow investors to diversify their portfolios by investing in all the companies in the index.

The relationship between the S&P 500 Index and the US economy is manifested in several ways. First, the index includes the largest and most representative companies in various sectors of the economy. Therefore, its performance and movement reflect conditions and trends in a wide range of sectors and industries. The index is an important indicator of investor confidence and expectations about economic conditions in the United States. Rising prices in the index indicate optimistic investor sentiment and belief in continued economic growth. This can stimulate investment, business growth, and demand for labor. As mentioned above, the index is also used as a benchmark for evaluating the performance of investment portfolios and funds. Many active and passive investors use the index as a benchmark to compare and evaluate the performance of their investments. If an investment portfolio outperforms the index, it may indicate successful asset management (Investopedia, 2023). The index considered by many analysts and economists to be one of the most important indicators of the health and prospects of the U.S. economy. Changes in the index can serve as an indicator of future economic trends, growth or decline. Analysts study the relationship between the index's movements and macroeconomic factors such as GDP, inflation, and unemployment to predict possible economic outcomes (Jareño, 2016).

The S&P 500 Index plays an important role in reflecting and measuring the performance of stocks and the U.S. economy as a whole. Its movements and changes reflect important aspects of economic activity and investor sentiment. This makes it an indispensable tool for analyzing and forecasting economic developments and the market (Dattatray, 2019).

Literature Review

Johannes W. Flume (2021) note that the stock exchange means the organizer of commodities, securities and labor-powered wholesale sales on the basis of supply and demand in the economy, as well as for the sale of financial and trading transactions to sellers and buyers place (Johannes, 2021). In Investopedia (2023) and IndiaCharts (2022) express the view that trading on the stock exchange is conducted according to certain rules and procedures, and all transactions are registered and monitored by the relevant regulatory authorities (SoFi, 2023).

Over time, trading mechanisms and institutions evolved, contributing to the development of international trade and laying the foundation for modern stock markets.

The growth of industry and the emergence of new companies created new investment opportunities and stimulated the development of the stock market. However, with this development came the need to establish regulatory measures and protect the interests of investors. In response to these needs, rules and regulations were introduced and specialized organizations responsible for the control and supervision of stock exchanges and companies were established. Such organizations play an important role in ensuring stability and creating confidence in the market by ensuring that trading rules are followed, investor rights are protected, and fraud is prevented (Stock Market History) (IndiaCharts, 2022).

Malkiel, B. G. (2015) express the view that today, the stock market continues to undergo active technological innovations that are changing the very process of conducting transactions and accessing information. Electronic trading, process automation and the development of online platforms have created opportunities for investors to trade stocks and other financial instruments with greater efficiency and convenience (Malkiel, 2015). Wiley, J. (2017) emphasize that now the exchange market is a complex system where various financial assets, such as stocks, bonds, commodities and derivatives, are traded (Wiley, 2017). Fatemeh Aramian (2021) conclude that it is an electronic platform where buyers and sellers make transactions based on supply and demand (Aramien, 2021).

Pisano, U. A., Martinuzzi, B., & Bruckner, B. (2012) draw attention to the fact that the stock market performs a number of economic functions: allows lenders and investors to invest their money in various financial assets, such as stocks and bonds (this increases the amount of financial resources available and encourages investment activity); plays an important role in providing information about the financial condition of companies and projects related to various financial assets (this reduces the cost of accessing such information, making it more accessible and allowing for more informed investment decisions); provides liquidity to holders of financial assets (owners of stocks and bonds can sell their investments in the market when they need funds or want to reallocate their investments. This creates the ability to quickly convert assets into cash); serves as a platform for the development and evolution of various methods of financing projects (it provides an opportunity for companies and organizations to raise capital to finance their projects through the issuance of stocks and bonds of various types and maturities) (Pisano, 2012).

Merritt B. Fox. (2021) note that the stock market plays a key role in stimulating economic growth and wealth creation, so accurate forecasts of its state are important for the overall stability and efficiency not only of financial markets, but also of national economies. The study and development of modeling methods, stock value forecasting is a practically significant task for all market participants and those who just want to enter the market, allowing to make informed decisions to managing an investment portfolio.

One of Henry's seminal works, "Stock Market Liberalization, Economic Reform and Stock Prices in Emerging Markets", emphasizes that stock markets play a crucial role in facilitating the relationship between savers and producers in society (Henry, P. B. (1997). Savers, who have accumulated a surplus of funds, seek to invest their savings in profitable and ambitious projects. On the other hand, producers, representing the productive sectors of the economy, need financial resources to fuel their activities and promote economic growth.

Mishkin, F. S., & Eakins, S. G. (2014) draw attention to the fact that stock markets act as intermediaries, allowing the transfer of funds from savers to producers. This process allows productive sectors to access the necessary capital for expansion and development. The productivity and functions of the stock market play an important role in redirecting funds from those who have excess resources to those who need them, thereby facilitating economic activity and development.

Metodology

Statistical methods include a large number of methods, such as methods of valuation theory, factor analysis, regression and correlation analysis, etc. With the help of these methods, investors can conduct a comprehensive statistical research of the financial market, make forecasts of market processes and, on the basis of these forecasts, make more reasonable investment decisions. However, working with such systems for forecasting short-term price movements, rapidly changing intraday information is associated with some difficulties both in the selection of an analysis method and in the interpretation of results. This seems to be a significant drawback, because the speed of forecasting intraday trading is very important (Malyshenko, 2014).

One of the simplest and most effective methods is the Autoregressive Integrated Moving Average (ARIMA): This model is one of the most common and simplest time series forecasting models. It is based on the assumption that the future values of a series depend on its past values and forecast errors. It is based on a combination of three main components: autoregression (AR), integration (I) and moving average (MA), as follows:

- Autoregression (AR): The model assumes that the future values of a time series depend on its past values. Autoregression uses the lags (previous values) of the series to predict its future values. Autoregression (AR order) determines the number of past values used in the model.
- Integration (I): Integration is used to ensure that the time series is stationary. If the original series is non-stationary (has trends or seasonality), it can be transformed into a stationary series by using the differences between successive observations. The integration order (I-order) determines the number of differences applied to the series.
- Moving Average (MA): The moving average assumes that the current value of the series depends on random forecast error at previous times. The model uses smoothing of the forecast errors to account for their effect on future values of the series. The MA order determines the number of past errors (Hyndman, 2018).

The next model that can be distinguished - GARCH (General Autoregressive Conditional Heteroscedastic) - is a model used to model and predict the volatility of time series, such as the prices of financial instruments, including the SP500 index. The GARCH model is based on the assumption that the variance of a series varies over time and depends on the previous values of the series. The model is widely used in financial econometrics and time series analysis to model and predict the volatility of financial data. It has advantages in modeling variability and accounting for variance structure in time series. However, like any model, GARCH has its limitations and requires proper parameter selection and estimation to achieve good results in predicting volatility (Sheeen, 2016).

Random Forest is a machine learning algorithm that combines multiple decision trees to perform classification and regression tasks. It uses randomness to select features and data samples, and combines tree predictions to improve model accuracy and stability. Random forest has high performance, the ability to estimate the importance of features, robustness to overlearning, and a wide range of applications in a variety of domains (Uzakariya, 2021).

LSTM (Long Short-Term Memory) is a tool for stock market forecasting and modeling. Unlike traditional models, it is able to handle long-term dependencies in time series and capture complex time patterns. It has a built-in ability to remember and forget information over time, allowing it to account for long-term trends and seasonal fluctuations in the market. LSTM also has the ability to model non-linear dependencies and adapt to changing market conditions. It can use different types of data, including stock prices, trading volumes, macroeconomic indicators, and other factors to make more accurate predictions. Numerous studies and publications demonstrate the successful application of this model in stock market forecasting and describe various methods and approaches to its use (Serafeim, 2020).

VAR (Vector Autoregression) is a model that is widely used for stock market forecasting and modeling. It allows you to analyze the relationships between multiple time series, taking into account the impact of one series on others. A VAR model is a system of simultaneous equations in which each variable depends on its past values and the past values of other variables. This allows for complex interactions and dependencies between various factors such as stock prices, trading volumes, market indicators, and economic performance. The VAR model has the ability to capture dynamics and long-term trends in time series and to predict future values based on past data. Its advantages include the ability to analyze historical relationships, estimate

impulse response, and perform scenario analysis. There are many papers in the financial econometrics literature and research studies that apply the VAR model to stock market forecasting and modeling, and describe methods for estimating and interpreting the results (Longmore, 2020).

XGBoost (Extreme Gradient Boosting) - Combines weak models, such as decision trees, to improve predictions. Works with different types of features, automatically selects important features, and is resistant to overfitting. It has several important advantages. First, it provides high speed and efficiency due to its optimized implementation of gradient binning, making it ideal for dealing with large amounts of financial data. Second, can handle both numeric and categorical attributes, allowing it to account for a variety of factors in stock market analysis, providing models with flexibility and accuracy. Third, the algorithm automatically determines the importance of the attributes and selects the most important ones, improving the quality of forecasts and simplifying the model by removing unnecessary attributes (Rahman, 2023). XGBoost has built-in regularization mechanisms that prevent model overlearning and provide more reliable stock market forecasts (Dat Tan Trinh, 2022).

Exponential smoothing is a time series forecasting method that uses a weighted average of past observations with decreasing weight as you move away in time. The basic idea of exponential smoothing is to give more weight to more recent observations and less weight as you move away from the current moment. This allows us to model the impact of newer data on predictions while taking into account the decreasing importance of older data.

Results

Dataset was taken from the Yahoo Finance website (Yahoo Finance, 2023), that contains historical data for the S&P 500 Index. This daily dataset covers the period from January 3, 1990 to May 31, 2023 and provides information about the opening price, the maximum and minimum price, the closing price and the adjusted closing price.

This case study will provide a basic understanding of the data structure and provide reasonably accurate future price predictions, which can be a great tool to minimize the risk of losing money in the market.

We will do our work using the Python programming language version 3.10.12. You should start by importing all the necessary components (Figure 1). These libraries and modules play a crucial role in data analysis, visualization, time series modeling, and evaluation of model performance on the dataset. They provide a wide range of functions and tools that simplify various aspects of data analysis and forecasting in the context of financial markets. They are also divided into sections, and you can see a description of them in the comments

```

1 !pip install pmdarima
2
3 import warnings
4 warnings.filterwarnings('ignore')
5
6 # Data manipulation and analysis libraries
7 import numpy as np
8 import pandas as pd
9
10 # Data visualization libraries
11 import matplotlib.pyplot as plt
12 import seaborn as sns
13
14 # Time series analysis libraries
15 from statsmodels.tsa.stattools import adfuller
16 from statsmodels.tsa.seasonal import seasonal_decompose
17 from statsmodels.tsa.arima.model import ARIMA
18 from pmdarima.arima import auto_arima
19
20 # Evaluation metrics libraries
21 from sklearn.metrics import mean_squared_error, mean_absolute_error

```

Figure 1. Import Libraries

Source: compiled by the author using the Python programming language version 3.10.12.

All necessary libraries have been downloaded, now you need to import and read the data set (Figure 2).

```
1 df = pd.read_csv("S&P 500 Historical Data.csv")
2 df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-01-03	1469.250000	1478.000000	1438.359985	1455.219971	1455.219971	931800000
1	2000-01-04	1455.219971	1455.219971	1397.430054	1399.420044	1399.420044	1009000000
2	2000-01-05	1399.420044	1413.270020	1377.680054	1402.109985	1402.109985	1085500000
3	2000-01-06	1402.109985	1411.900024	1392.099976	1403.449951	1403.449951	1092300000
4	2000-01-07	1403.449951	1441.469971	1400.729980	1441.469971	1441.469971	1225200000
...
5885	2023-05-24	4132.959961	4132.959961	4103.979980	4115.240234	4115.240234	3739160000
5886	2023-05-25	4155.709961	4165.740234	4129.729980	4151.279785	4151.279785	4147760000
5887	2023-05-26	4156.160156	4212.870117	4156.160156	4205.450195	4205.450195	3715460000
5888	2023-05-30	4226.709961	4231.100098	4192.180176	4205.520020	4205.520020	4228510000
5889	2023-05-31	4190.740234	4195.439941	4166.149902	4179.830078	4179.830078	5980670000

5890 rows x 7 columns

Figure 2. Import S&P 500 Historical Data

Source: compiled by the author.

And first, we need to check all the information about the data, check the data types and also check for missing data in the set (Figure 3).

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5890 entries, 0 to 5889
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        5890 non-null   object
1   Open        5890 non-null   float64
2   High        5890 non-null   float64
3   Low         5890 non-null   float64
4   Close       5890 non-null   float64
5   Adj Close   5890 non-null   float64
6   Volume      5890 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 322.2+ KB
```

```
1 # Check for null values
2 null_values = df.isnull().sum()
3 print(null_values)
```

```
Date          0
Open          0
High          0
Low           0
Close         0
Adj Close     0
Volume        0
dtype: int64
```

Figure 3. Dataset information and check for null values

Source: compiled by the author.

In this paper, we only need the closing price for the entire study. To focus specifically on the closing price, we can extract the 'Close' column from the dataset and store it in a new variable called df_close (Figure 4). This will allow us to perform further analysis and calculations on the closing prices only.

```
1 df_close = df['Close']
```

Figure 4. Extract the 'Close' column from the dataset

Source: compiled by the author

For further successful data visualization, let's set the 'Date' column as an index, it gives us the opportunity to replace the serial number of the rows with a specific date of observation (Figure 5).

```
1 df['Date'] = pd.to_datetime(df['Date'])
2 df = df.set_index(df['Date']).sort_index() # setting date feature as our index
3 print(df.shape)
4 df.sample(5)
```

(5890, 7)

	Date	Open	High	Low	Close	Adj Close	Volume
	2022-05-19	3899.000000	3945.959961	3876.580078	3900.790039	3900.790039	5113550000
	2016-05-31	2100.129883	2103.479980	2088.659912	2096.949951	2096.949951	4514410000
	2014-06-03	1923.069946	1925.069946	1918.790039	1924.239990	1924.239990	2867180000
	2001-01-23	1342.900024	1362.900024	1339.630005	1360.400024	1360.400024	1232600000
	2000-01-03	1469.250000	1478.000000	1438.359985	1455.219971	1455.219971	931800000

Figure 5. Setting the 'Date' column as an index

Source: compiled by the author.

Now we can look at the full chart of our price - it will create a line chart of closing prices, where the x-axis represents the dates and the y-axis represents the corresponding closing values (Figure 6).

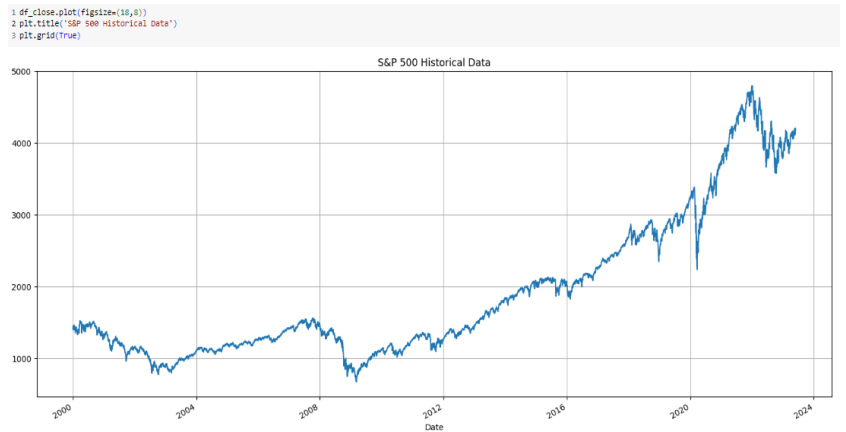


Figure 6. Plot S&P 500 Historical Data

Source: compiled by the author.

Next, we'll create a Kernel Density Estimation (KDE) graph that estimates the probability density function of the data, providing insight into the distribution and shape of the data. This KDE chart allows you to visualize the distribution of closing prices. The resulting curve provides an estimate of the probability density function, with higher peaks indicating areas of higher density and lower troughs indicating areas of lower density. The shading below the curve provides a visual representation of the estimated probability density function (Figure 7).

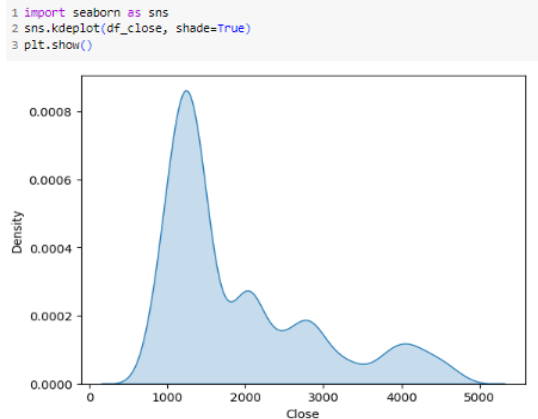


Figure 7. Kernel Density Estimation graph

Source: compiled by the author.

Kernel density estimation (KDE) is a method used to estimate the probability density function of a random variable from a given data set. The graph can provide insight into the central trend of closing prices. The location of the highest peak on the KDE chart corresponds to the mode of the distribution that represents the most frequent closing price. This can be useful for traders in identifying potential support or resistance levels - from this point of view we can identify four support levels, the main one being around 1200. It is also important to note that the KDE chart can help identify potential outliers in the closing prices. Outliers are data points that deviate significantly from the overall distribution pattern. These outliers may represent important events or anomalies that have affected the closing prices - of which we do not observe any .

We can move on to more basic things like the Dickey-Fuller test for stationarity. Data are stationary if they have no trend or seasonal effects. And if the data is non-stationary, we need to convert it to stationary before we can fit it to an ARIMA model. But before we do that, construct a rolling mean and a rolling standard deviation, which are statistical metrics that help to estimate the mean and the dispersion of the values in the time series in a given window (Figure 8).

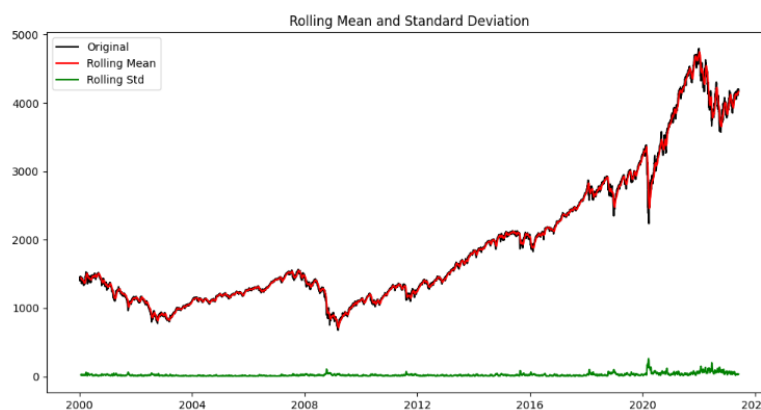


Figure 8. Rolling Mean and Standard Deviation

Source: compiled by the author.

The moving statistic chart created in the code (Fig. 2.8) helps us to visually assess the stationarity of the time series before applying the Dickey-Fuller test. Drawing these curves allows us to visually assess whether the data has a trend (change in mean value over time) or the presence of a changing variance (dispersion of values). If the graph shows clear trends or changes in the variance of the data, this may indicate non-stationarity of the time series. This function `def test_stationarity` (Figure 9) is useful for analyzing time series and their stationarity. Stationary time series have a constant mean and variance, which makes forecasting and modeling easier. If a time series is not stationary, additional transformations or modeling may be required to achieve stationarity and more accurate analysis.

```

1 def test_stationarity(timeseries):
2     # Definition of rolling statistics
3     rolmean = timeseries.rolling(window=12).mean()
4     rolstd = timeseries.rolling(window=12).std()
5
6     # Create a graph of moving statistics
7     plt.figure(figsize=(12, 6))
8     plt.plot(timeseries, color='black', label='Original')
9     plt.plot(rolmean, color='red', label='Rolling Mean')
10    plt.plot(rolstd, color='green', label='Rolling std')
11    plt.legend(loc='best')
12    plt.title('Rolling Mean and Standard Deviation')
13    plt.show()
14
15    # Test for stationarity with the Dickey-Fuller test
16    print("Results of Dickey-Fuller test:")
17    adft = adfuller(timeseries, autolag='AIC')
18    output = pd.Series(adft[0:4], index=['Test Statistics', 'p-value', 'No. of lags used', 'Number of observations used'])
19
20    for key, value in adft[4].items():
21        output['Critical value (%)' % key] = value
22
23    print(output)
24
25 test_stationarity(df_close)

```

Figure 9. Def test_stationarity with definition of rolling statistics

Source: compiled by the author.

And also, in addition to the graph with the moving statistics, it gives us the results of the test - Augmented Dickey Fuller test - the unit root test Let us first see the formula for the Dickey Fuller test which is the origin of the Augmented Dickey Fuller test, and that is (1).

$$Y_t = c + \beta_t + aY_{t-1} + \phi\Delta Y_{t-1} + e_t \quad (1)$$

where,

Y_t = value in the time series at time t or lag of 1 time series.

$\phi\Delta Y_{t-1}$ = first difference of the series at time (t-1)

The formula for Augmented Dickey Fuller test, and it goes as follows (2.2):

$$Y_t = c + \beta t + aY_{t-1} + \phi\Delta Y_{t-1} + \phi_2\Delta Y_{t-2} \dots + \phi_p\Delta Y_{t-p} \quad (2)$$

The formula for ADF is the same equation as the DF with the only difference being the addition of differencing terms representing a larger time series. Fundamentally, it has a similar null hypothesis as the unit root test. That is, the coefficient of $Y(t-1)$ is 1, implying the presence of a unit root. If not rejected, the time series is taken to be non-stationary. If null hypothesis is rejected, then Test statistic < Critical Value and p-value < 0.05, the time series is stationary (Wooldridge, 2019), (Yang, 2022). The result of the function (Figure 9) is available to see in (Figure 10).

```

Results of Dickey-Fuller test:
Test Statistics                1.015948
p-value                        0.994431
No. of lags used              32.000000
Number of observations used    5857.000000
Critical value (1%)           -3.431467
Critical value (5%)           -2.862034
Critical value (10%)          -2.567033
dtype: float64

```

Figure 10. Results of Augmented Dickey Fuller test

Source: compiled by the author.

Analysis of test results (Figure 10):

- Test Statistics has a positive value of 1.015948. Compared with the critical values, this indicates that the test statistic is far from zero and is not negative enough to confirm the stationarity of the series.
- The p-value is 0.994431, which is a high value close to 1. This means that there is a high probability of obtaining such or more extreme results even if the null hypothesis of non-stationarity of the series is true. Therefore, we do not have sufficient evidence to reject the null hypothesis.
- Critical values) are shown as -3.431467 (1%), -2.862034 (5%), -2.567033 (10%). They are threshold values compared to the test statistic. If the test statistic is less than the critical value, the null hypothesis of non-stationarity is rejected. In this case, the value of the test statistic is not low enough compared to the critical values, which confirms the lack of stationarity in the series.
- In summary, based on the results obtained, we can conclude that the time series is not stationary, since we have not rejected the null hypothesis of non-stationarity. This may indicate the presence of a trend, seasonal fluctuations or other systematic changes in the data.
- The next logical step is to separate the seasonality from the trend before analyzing the time series. Such an approach will lead to stagnation of the resulting series. (Figure 11).

```

1 # Apply seasonal decomposition
2 result = seasonal_decompose(df_close, model='multiplicative', period=365)
3
4 # Visualization of decomposition results
5 with plt.rc_context({'figure.figsize': (18, 8)}):
6     fig = result.plot()
7
8 # Differentiation to achieve stationarity
9 df_close_diff = df_close.diff().dropna()

```

Figure 11. Seasonal decomposition

Source: compiled by the author.

The result of the code (Figure 11) is available to see in (Figure 12).

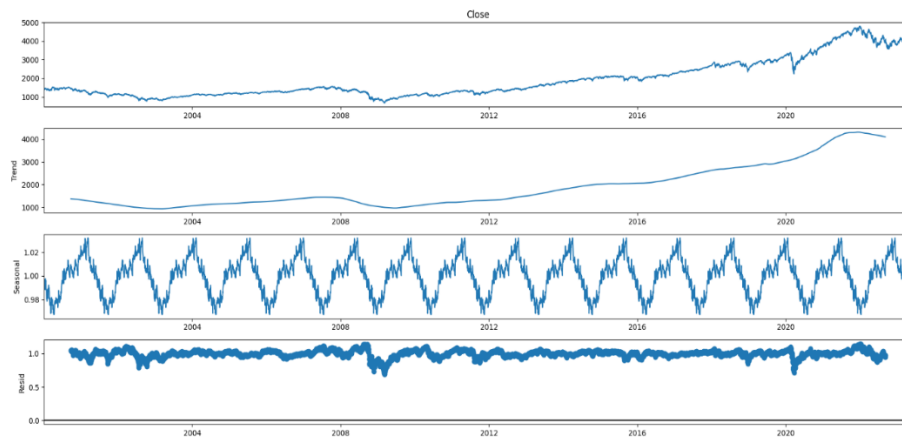


Figure 12. Results of seasonal decomposition

Source: compiled by the author.

Also consider the second version of stationarity - to reduce the magnitude of the values and the increasing trend in the series, we first take a log of the series. Then, after obtaining the logarithm of the series, we compute the rolling average of the series. A rolling average is calculated by taking data from the previous 12 months and calculating an average consumption value at each subsequent point in the series. The following code is used to smooth the time series and analyze its variability. The logarithmic transformation helps to reduce non-stationarity and smooth fluctuations in the data, and calculating the moving average and standard deviation allows you to assess the overall trend and variability of the series (Figure 13).

```

1 # Set the size of the chart shape
2 plt.figure(figsize=(10, 6))
3
4 # Apply a logarithmic transformation to the time series
5 df_log = np.log(df_close)
6
7 # Calculate moving average and standard deviation
8 window = 12
9 moving_avg = df_log.rolling(window).mean()
10 std_dev = df_log.rolling(window).std()
11
12 # Plot moving average and standard deviation charts
13 plt.plot(std_dev, color="black", label="Standard Deviation")
14 plt.plot(moving_avg, color="red", label="Mean")
15
16 # Chart legend and title settings
17 plt.legend(loc="best")
18 plt.title("Moving Average and Standard Deviation")
19
20 plt.show()

```

Figure 13. Moving average and standard deviation

Source: compiled by the author.

The plot of the code (Figure 13) is available to see in (Figure 14).

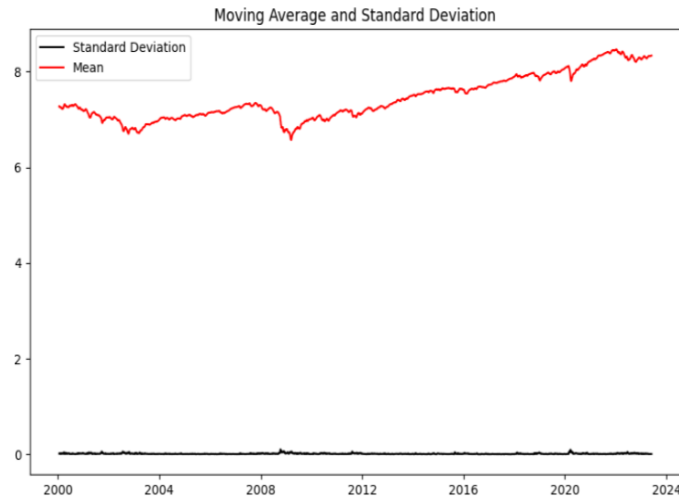


Figure 14. Plot of Moving Average and Standard Deviation

Source: compiled by the author.

The process described in the code (Figure 15) is called “trend removal” or “detrending” a time series. It is an important step in time series analysis and can help reveal hidden features and patterns in the data. Trend removal is performed by subtracting the moving average from the original time series. This highlights shorter-term fluctuations, such as business cycles and seasonal patterns, and makes it easier to analyze these components.

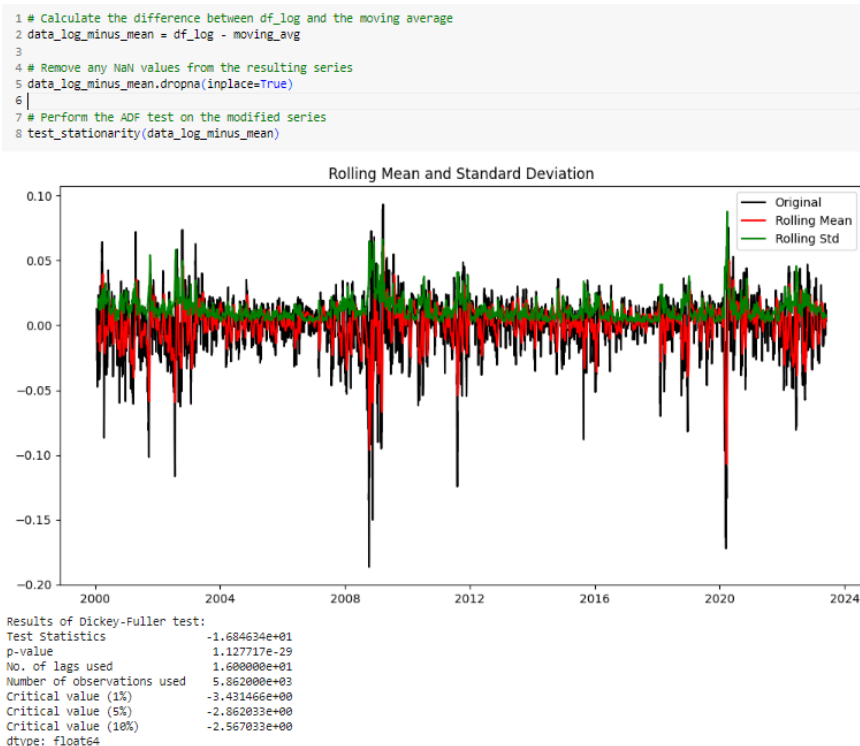


Figure 15. Calculate the difference between df_log and the moving average and ADF results

Source: compiled by the author.

And also from (Figure 15) available to see the updated result of the ADF test. Based on these results we can draw the following conclusions:

- Test Statistic (-1.684634e+01) is less than the critical values for all significance levels (-3.431466e+00, -

2.862033e+00 and -2.567033e+00). This indicates that we can reject the null hypothesis of unit root and accept the alternative hypothesis of stationarity of the time series.

- P-value (1.127717e-29) is very close to zero, which also confirms the statistical significance of the test results. Usually, if the p-value is smaller than the selected significance level (Naushad, 2020), we can reject the null hypothesis and conclude that the series is stationary. In this case, the p-value is much less than 0.05, which confirms the stationarity of the series.
- Thus, based on the results of the Dickey-Fuller test with a very low p-value and test statistics lower than the critical values, we can conclude that the time series is stationary. This ensures the stability of the model and allows the use of past data to accurately predict future values.
- The ARIMA model is one of the most popular models for making short-term forecasts. To describe this model, three groups of parameters are used: p, d, and q are nonnegative integers that characterize the order of the model parts (autoregressive, integrated, and moving average, respectively). (Chumachenko, 2020)
- Parameter p (AR - Autoregressive): Indicates the number of previous values of the time series used to predict the current value. If p=1, the model uses only one previous value. If p=2, the model uses two previous values, and so on. A larger value of p means that the model uses more previous values for prediction.
- The parameter d (I - Integration): It determines the number of differentiations needed to achieve stationarity of the series. Differentiations help to remove trends and seasonality in the series. If d=0, the series is considered stationary. If d=1, then the model uses the first difference value of the series. If d=2, the model applies the second difference value of the series, and so on.
- Parameter q (MA - Moving Average): It indicates the number of previous prediction errors used to predict the current value. If q=1, the model uses only the previous error. If q=2, the model considers two previous errors, and so on. A larger value of q means that the model uses more previous errors for prediction.

The parameters p, d, and q together define the structure of the ARIMA model. For example, an ARIMA(1, 1, 1) model means that an autoregression of order 1, a differentiation, and a moving average of order 1 are used. The choice of the optimal values of the parameters p, d and q can be based on the analysis of the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the time series, as well as on the use of statistical criteria and model evaluation methods (Hyndman, 2018). In our work we will use the function of automatic selection of parameters (Figure 16).

```

1 # Create and train an AutoARIMA model
2 model = auto_arma(df_log, start_p=1, start_q=1,
3                 max_p=3, max_q=3, m=1,
4                 start_P=0, seasonal=False,
5                 d=None, D=None, trace=True,
6                 error_action='ignore',
7                 suppress_warnings=True,
8                 stepwise=True)
9
10 # Obtaining optimal model parameters
11 order = model.order

```

Figure 16. Create and train an AutoARIMA model

Source: compiled by the author.

The `auto_arma` function (Figure 16) performs automatic model parameter tuning and is a convenient tool that allows you to automatically select the optimal ARIMA model parameters based on statistical analysis and heuristic methods. It facilitates the process of model tuning with a simple function rather than searching for p, d and q parameters separately. The result is shown in (Figure 17).

```

Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=-34976.643, Time=2.63 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=-34914.679, Time=0.50 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=-34976.612, Time=1.01 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=-34978.337, Time=5.80 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=-34915.465, Time=0.31 sec
ARIMA(0,1,2)(0,0,0)[0] intercept : AIC=-34976.660, Time=2.26 sec
ARIMA(1,1,2)(0,0,0)[0] intercept : AIC=-34974.731, Time=2.53 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=-34978.785, Time=0.95 sec
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=-34977.066, Time=2.11 sec
ARIMA(0,1,2)(0,0,0)[0] intercept : AIC=-34977.083, Time=1.82 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=-34977.104, Time=0.68 sec
ARIMA(1,1,2)(0,0,0)[0] intercept : AIC=-34975.161, Time=2.12 sec

Best model: ARIMA(0,1,1)(0,0,0)[0]
Total fit time: 22.757 seconds

```

Figure 17. Results of AutoARIMA model

Source: compiled by the author.

The results (Figure 17) are presented as a table, where each row corresponds to the ARIMA model with certain parameter values, and the columns show the following information:

- ARIMA(p, d, q)(P, D, Q)[m]: ARIMA model parameters, where p, d, q are order of autoregression, integration and moving average respectively, P, D, Q are order of seasonal autoregression, integration and moving average respectively, m is seasonal period (in our case have zero values due to choice of model and specifying seasonal=False parameter).
- AIC: Akaike Information Criterion (AIC), which is a measure of the relative quality of the model, where a smaller value of AIC indicates a better model.

Based on the results, the best ARIMA model is ARIMA(0,1,1)(0,0,0)[0], which has the lowest AIC value of -34978.785.

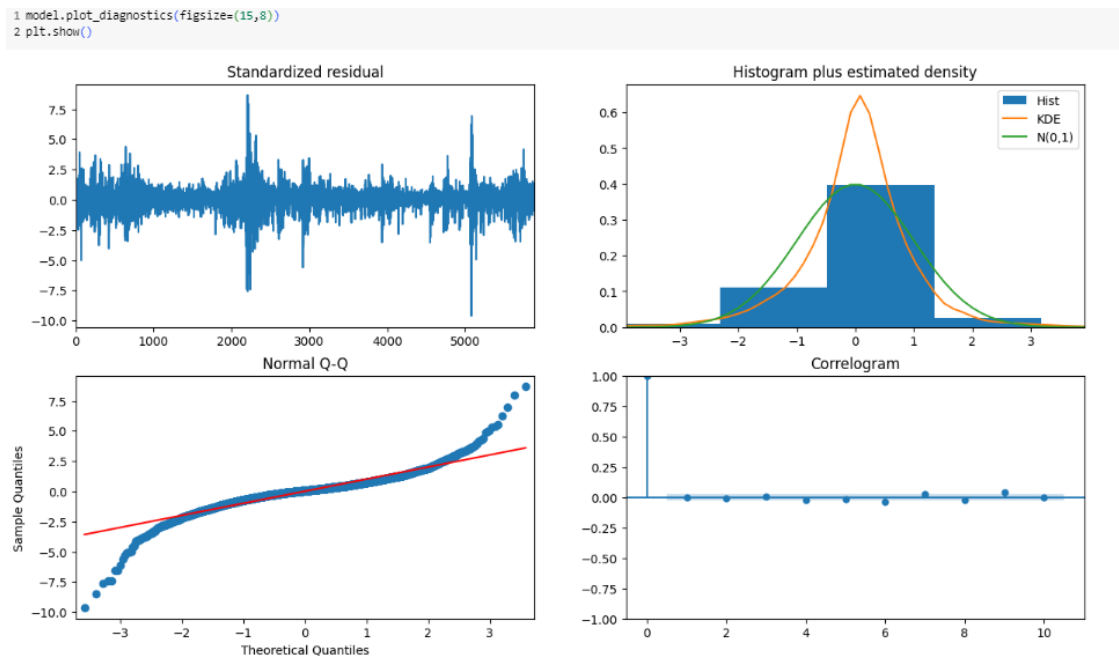


Figure 18. Model plot diagnostics

Source: compiled by the author.

The result of the (Figure 18) shows to us:

- Top left: The residual errors, which are the difference between the predicted values and the actual values, are plotted. If the variance of the residuals is homogeneous and they fluctuate around a mean of zero, this

indicates that the model assumptions of constant error variance are met.

- Top right: The density plot shows the distribution of the residual errors. If the distribution appears to be approximately normal and centered around zero (mean of zero), this indicates that the model's assumption of normally distributed errors is reasonable.
- Bottom left: A red line is mentioned that should fit all points perfectly. This may refer to a plot of predicted versus actual values. If the points closely match the red line, it indicates that the model predictions are accurate and unbiased. Any significant deviation would indicate a possible systematic error or asymmetric error distribution.
- Bottom right: The correlogram, also known as the ACF (autocorrelation function) plot, shows the autocorrelation of the residual errors. If the residual errors are not autocorrelated, it means that there is no significant pattern or relationship between the errors at different delays. This is desirable because it indicates that the model is adequately capturing the temporal dependencies in the data.

Next, we need to divide our performance into test and training data at a ratio of 15:85 (Figure 19).

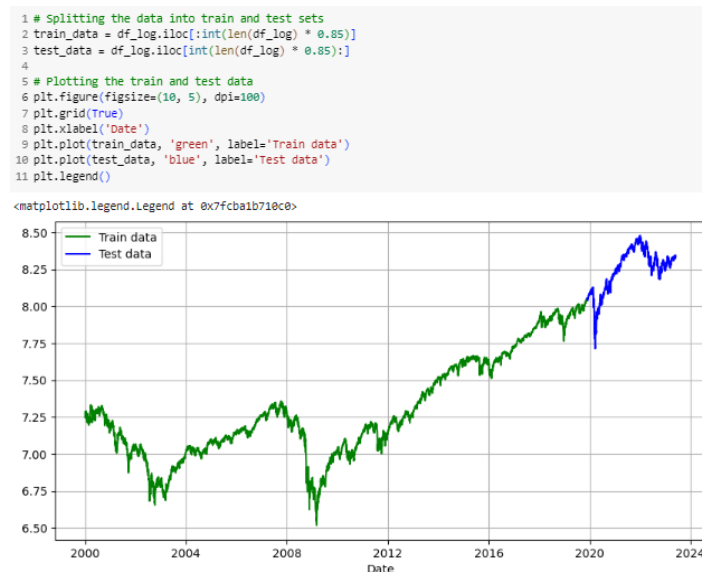


Figure 19. Splitting the data into train and test sets

Source: compiled by the author.

As a result, the Auto ARIMA model assigns the values 0, 1, and 1 to p, d, and q, respectively will input these parameters to our model (Figure 20).

```

1 model = ARIMA(train_data, order=(0,1,1))
2 fitted = model.fit()
3 print(fitted.summary())

```

Figure 20. Building and training an ARIMA model

Source: compiled by the author.

The result of the model (Figure 20) is available to see in (Figure 21).

ARIMA Results						
Dep. Variable:	Close	No. Observations:	5006			
Model:	ARIMA(0, 1, 1)	Log Likelihood	15082.405			
Date:	Fri, 9 Jun 2023	AIC	-30160.811			
Time:	15:25:27	BIC	-30147.774			
Sample:	0	HQIC	-30156.242			
	- 5006					
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ma.L1	-0.0777	0.008	-9.649	0.000	-0.093	-0.062
sigma2	0.0001	1.24e-06	113.645	0.000	0.000	0.000
Ljung-Box (L1) (Q):			0.01	Jarque-Bera (JB):	15109.89	
Prob(Q):			0.94	Prob(JB):	0.00	
Heteroskedasticity (H):			0.51	Skew:	-0.32	
Prob(H) (two-sided):			0.00	Kurtosis:	11.49	

Figure 21. ARIMA results

Source: compiled by the author.

Based on the results of the ARIMA methods provided in (Figure 21), we can make an express test (Smigel, 2021):

- The coefficient ma.L1 is -0.0777, which means that the model uses a lag of the difference series to predict public values. The negative sign of the hazard of detection between the variances and the current value of the relationship.
- The sigma2 is 0.0001. This is a very small value that the model is good at predicting the estimated time series.
- The Ljung-Box criterion (Q) has a value of 0.01 and the p-value (Prob(Q)) is 0.94. This indicates that the autocorrelations of the residuals in the first lag are not significant.
- The value of heteroskedasticity (H) is 0.51, which indicates an increase in the heteroskedasticity of the signs.

Thus, we can say that the ARIMA (0, 1, 1) model gives good results because it has a significant factor and a low variance of the residuals and we can proceed with the forecast (Figure 22).

```

1 # Forecast
2 forecast = fitted.get_forecast(steps=884, alpha=0.05)
3
4 # Get forecast, standard errors and confidence intervals
5 fc_series = forecast.predicted_mean
6 se_series = forecast.se_mean
7 lower_series = forecast.conf_int().iloc[:, 0]
8 upper_series = forecast.conf_int().iloc[:, 1]

```

Figure 22. Forecasting with a trained ARIMA model and obtaining predictive values

Source: compiled by the author.

Predictions can be used to predict future values of a time series based on available historical data. The standard errors returned by the `se_mean` method allow you to estimate the uncertainty or scatter of the predicted values. The smaller the standard error, the more accurate and reliable the predictions will be. Confidence intervals obtained with the `conf_int` method provide information about the likely range in which the future values of the series will lie. Confidence intervals help assess the uncertainty of predictions and can be used to make more informed decisions based on the probability that future values will fall within a certain range. The result of the forecasting is available to see in Figure 23.


```

1 print("Forecast:\n", fc_series)
2 print("Standard Errors:\n", se_series)
3 print("Confidence Intervals:\n", lower_series, upper_series)

```

Forecast:

5006	8.042497
5007	8.042538
5008	8.042423
5009	8.041701
5010	8.042268
...	
5885	8.175310
5886	8.175876
5887	8.176458
5888	8.175910
5889	8.175348

Name: predicted_mean, Length: 884, dtype: float64

Standard Errors:

5006	0.011890
5007	0.016126
5008	0.019461
5009	0.022303
5010	0.024821
...	
5885	0.350065
5886	0.350294
5887	0.350523
5888	0.350751
5889	0.350979

Name: var_pred_mean, Length: 884, dtype: float64

Confidence Intervals:

5006	8.019193
5007	8.010931
5008	8.004280
5009	7.997989
5010	7.993619
...	
5885	7.489194
5886	7.489313
5887	7.489446
5888	7.488451
5889	7.487441

Name: lower Close, Length: 884, dtype: float64

5006	8.065801
5007	8.074144
5008	8.080566
5009	8.085414
5010	8.090916
...	
5885	8.861425
5886	8.862440
5887	8.863469
5888	8.863370
5889	8.863254

Name: upper Close, Length: 884, dtype: float64

Figure 23. Output of forecast values, standard errors and confidence intervals

Source: compiled by the author.

The plotting of the forecast data is available to see in (Figure 24).

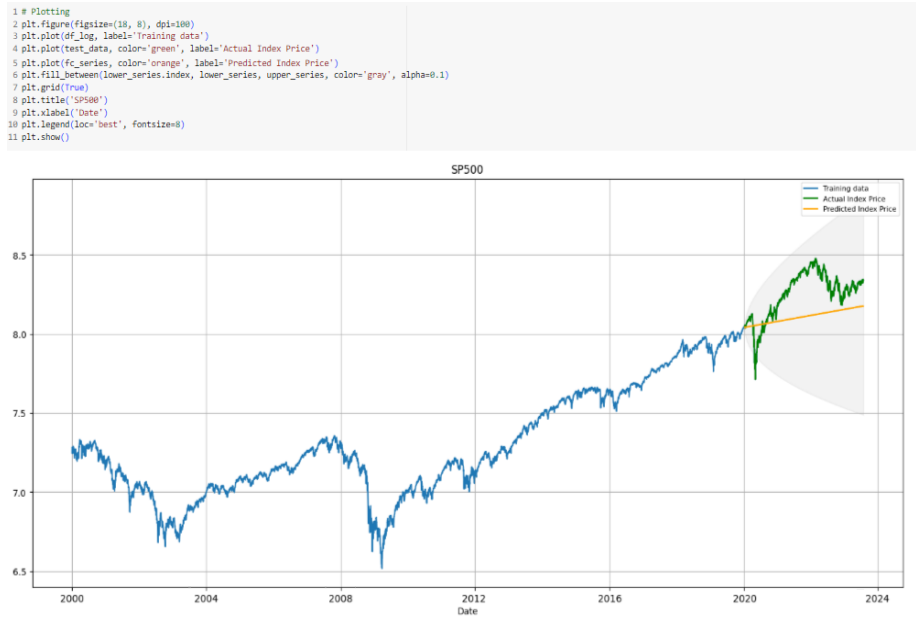


Figure 24. Plot with Predicted Index price

Source: compiled by the author.

When we evaluate the results of the forecast plot (Fig. 2.23), we can mention that it looks realistic and close to the test data. The only thing that stands out is March 2020 period, when we lost about 30% of the index value in one week, but we all know what an out-of-state situation is due to the pandemic and the general panic in the market. Such situations are extremely difficult to predict and such collapses are only possible by analyzing the news background. Similar crashes can also happen as a result of force majeure, for example, the recent incident that happened to Equifax, the large-scale cyber-attack that compromised the personal information of approximately 147 million people. As a result of the attack, Equifax's stock price plummeted and the company suffered significant financial losses - the stock price continued to decline in the following weeks, with a total loss in value of approximately 35% [45]. But it is only relevant in the context of one specific company - this is another advantage of working with indices - such events in one company do not have a critical impact on the index as a whole. And our predictable data clearly show the data to which we returned after the correction, which gives us the opportunity to make long-term forecasts, in our case the forecast was for the next 884 steps/day. We also need to be sure that we can call our model an accurate one. There are many metrics to evaluate the quality of a model. These (Figure 25) metrics allow you to evaluate how accurately and reliably the model is able to predict the values of the S&P 500 Index under study.

```

1 # Calculation of MAE (Mean Absolute Error)
2 mae = mean_absolute_error(test_data, fc_series)
3 # Calculation of MSE (Mean Squared Error)
4 mse = mean_squared_error(test_data, fc_series)
5 # Calculation of RMSE (Root Mean Squared Error)
6 rmse = np.sqrt(mse)
7 # Calculation of MAPE (Mean Absolute Percentage Error)
8 mape = np.mean(np.abs((test_data - fc_series) / test_data))
9
10 # Printing the results
11 print("MAE:", mae)
12 print("MSE:", mse)
13 print("RMSE:", rmse)
14 print("MAPE:", mape)
    
```

MAE: 0.16072026683536186
 MSE: 0.0347573081524168
 RMSE: 0.1864331197840577
 MAPE: 0.01934220013333761

Figure 25. Calculation of ARIMA model evaluation metrics

Source: compiled by the author.

Interpretation of the results (Nochovny, 2020), (Dyakonov, 2022), (Vandeput, 2019):

- MAE (Mean Absolute Error): The average difference between the predicted values and the actual values is approximately 0.161. It represents the average magnitude of the errors without considering their direction. On average, the forecasted values deviate from the actual values by around 0.161 units.
- MSE (Mean Squared Error): The average of the squared differences between the predicted values and the actual values is approximately 0.035. It measures the average squared magnitude of the errors, giving more weight to larger errors. A smaller MSE indicates a better fit between the predicted and actual values.
- RMSE (Root Mean Squared Error): The square root of MSE is approximately 0.186. It provides a measure of the average magnitude of the errors in the same unit as the original data. In this case, the RMSE suggests that, on average, the forecasted values deviate from the actual values by around 0.186 units.
- MAPE (Mean Absolute Percentage Error): The average percentage difference between the predicted values and the actual values is approximately 0.019. It measures the average magnitude of the errors as a percentage of the actual values. In other words, on average, the forecasted values deviate from the actual values by around 1.9%.

Conclusion

The growing importance of stock price forecasting has attracted considerable attention from industry experts and investors. Analyzing stock market trends is challenging due to the inherently noisy environment and significant volatility associated with market trends. The complexity of stock prices involves several factors, including quarterly earnings reports, market news, and changing investor behavior. Traders rely on a variety of technical indicators derived from daily stock market data. Despite the use of these indicators to analyze stock returns, accurately predicting daily and weekly market trends remains a challenge. Accurately predicting stock trends is a fascinating and challenging task in an ever-changing industrial world. Several aspects that influence stock market behavior are both non-economic and economic factors that are taken into consideration. Thus, stock market forecasting is considered a major challenge for increasing production. Traditional methods show that stock market returns are predicted based on past stock returns, other financial variables, and macroeconomics. Predicting stock market returns has led investors to investigate the reasons for predictability. Forecasting stock market trends is a complex process because it is influenced by many aspects, including traders' expectations, financial circumstances, administrative events, and certain factors related to market trends. Moreover, the stock price list is usually dynamic, complex, noisy, non-parametric and non-linear in nature. Financial time series forecasting becomes problematic due to certain complex characteristics such as volatility, irregularity, noise and changing trends.

Our choice of the ARIMA model in this research is based on the following considerations:

- Simplicity and interpretability: it is relatively easy to use and understand. It has a set of parameters, such as autoregression orders (p), difference (d), and moving average (q), that can be chosen based on data analysis and statistical metrics. This makes the model accessible to a wide range of users.
- Flexible model specification: Models are very flexible and can be customized to simulate different types of time series. This versatility is useful when working with multiple time series, as one type can be applied to different data sets.
- Time Dependency Accounting: The model accounts for time dependencies in the data, taking into account previous values in the series. This allows for trends, cyclicity, and seasonality in stock market time series. The model can capture long and short term dependencies, making it effective for forecasting financial time series.
- Suitable for all datasets: Can be trained on relatively small datasets due to fewer parameter requirements compared to neural networks or deep learning models. This makes them suitable when working with limited data availability.
- Robust performance: typically provide robust performance comparable to other time series statistical methods. While they may not always be the most efficient models, they provide consistent and reliable results, making them a good choice when time is limited for extensive experimentation.
- Prevalence and Availability: This is one of the most common time series forecasting models. It is well

studied in the literature and has extensive support in statistical packages and software tools. This makes the model accessible and usable in practical stock market forecasting tasks.

- Proven efficiency: The model has proven efficiency in forecasting time series, including financial data. Numerous studies and practical applications show that the model can be an effective tool for stock market forecasting.
- These advantages contribute to the attractiveness and usefulness of ARIMA models in forecasting and analyzing time series data, including our selected stock market index, the S&P 500.
- This article develops an ARIMA model for forecasting and analysis of time series data of the S&P 500 stock market index:
- EDA and Data Cleaning/Validation: Perform exploratory data analysis to understand the characteristics of the time series.
- Determine Moving Statistics: Compute moving statistics, such as moving averages or moving standard deviations, to identify trends and seasonality.
- Test for stationarity: Apply a stationarity test, such as the Augmented Dickey-Fuller test, to ensure that the time series is stationary. If the time series is not stationary, we will perform additional manipulations to achieve stationarity.
- Apply seasonal decomposition: Apply the seasonal decomposition method to decompose the time series into its components: trend, seasonality, and residuals. This will allow us to better understand the contribution of each component to the overall index dynamics.
- Applying the logarithmic transformation: Apply the logarithmic transformation to the time series to smooth the extreme values and reduce their impact on the model.
- Finding the optimal model parameters: Search for the optimal parameters for the ARIMA model - this will allow us to build the most accurate and appropriate time series model.
- Implement an ARIMA model to predict the S&P 500 index: Divide the data into training and test data and implement the ARIMA model using the optimal parameters to predict the future price.
- Analyzing the Results and Checking the Accuracy of the Model: Analyze the results of the analysis and compare the predicted values with the actual data to evaluate the accuracy and reliability of our model. This will allow us to draw conclusions about the applicability of the model for predicting the future dynamics of the index.

References

1. Johannes W. Flume. (2021). Constructing the Stock Exchange: On the Rise and Fall of an Iconic Place of Capitalism. [\[Link\]](#)
2. Investopedia. (2023). Stock Market. Investopedia. [\[Link\]](#)
3. SoFi. (2023). History of the Stock Market. [\[Link\]](#)
4. IndiaCharts. (2022). Stock Market History. [\[Link\]](#)
5. Malkiel, B. G. (2015). A Random Walk Down Wall Street: The Time-Tested Strategy for Successful Investing. W. W. Norton & Company. [\[Link\]](#)
6. Wiley, J. (2017). The New Stock Market: Law, Economics, and Policy. John Wiley & Sons. [\[Link\]](#)
7. Fatemeh Aramian. (2021). Off-exchange Trading in Modern Equity Markets. [\[Link\]](#)
8. Henry, P. B. (1997). Stock market liberalization, economic reform, and emerging market equity prices. *Journal of Finance*, 52(3), 1227-1246. [\[Link\]](#)
9. Mishkin, F. S., & Eakins, S. G. (2014). Financial Markets and Institutions. Pearson; 8th edition. [\[Google Scholar\]](#)

10. Pisano, U. A., Martinuzzi, B., & Bruckner, B. (2012). The Financial Sector and Sustainable Development: Logics, principles, and actors. ESDN Quarterly Report. [\[Link\]](#)
11. Author(s). (2016). The Impact of Stock Market Performance upon Economic Growth. [\[Link\]](#)
12. Merritt B. Fox. (2021). The Social Functions of the Stock Market. [\[Link\]](#)
13. Justin Kuepper. (2022). Major World Stock Market Indexes. The Balance. [\[Link\]](#)
14. John Egan. (2023). Dow vs. Nasdaq vs. S&P 500: What's the Difference? [\[Link\]](#)
15. Investopedia. (2023). Composite Index: Definition, Types, and Examples. [\[Link\]](#)
16. Ellis, C. D. (2016). The Index Revolution: Why Investors Should Join It Now. Foreword by B. G. Malkiel. Wiley; 1st edition. [\[Google Scholar\]](#)
17. Ganeshwaran Kana. (2022). Why the Stock Market Index Is Important for You. [\[Link\]](#)
18. Josef Novotný, Iveta Jaklová. (2022). The Importance of Global Financial Indices for Investors in Alternative Investment. [\[Link\]](#)
19. S&P Global. (2023). S&P 500 Brochure. [\[Link\]](#)
20. S&P Global. (2023). S&P 500 - Overview. [\[Link\]](#)
21. S&P Dow Jones Indices. (2023). S&P 500 Brochure, page 4. [\[Link\]](#)
22. Corporate Finance Institute. (2023). S&P 500 Index. [\[Link\]](#)
23. Author(s). (2021). Title of the article. Complexity. [\[Link\]](#)
24. Investopedia. (2023.). How the Stock Market Affects the Economy. <https://www.investopedia.com/how-stock-market-affects-economy-5296138/>
25. Francisco Jareño, (2016). US Stock Market and Macroeconomic Factors. [\[Link\]](#)
26. Dattatray P. Gandhmal. (2019). Systematic analysis and review of stock market prediction techniques" *Computer Science Review*, 36. [\[Link\]](#)
27. Malyshenko, K. A., Anashkina, M. V. (2014). Use of Neural Networks for the Purposes of Forecasting the Stock Market, *Efficient Economics*. [\[Link\]](#)
28. Hyndman, R. J. (2018). Forecasting: Principles and Practice ARIMA Models. [\[Link\]](#)
29. Sheereen Fauzel. (2016). A Generalized Autoregressive Conditional Heteroscedastic Approach for the Assessment of Weak-form Efficiency and Seasonality Effect: Evidence from Mauritius. [\[Link\]](#)
30. Uzakariya, M. (2021). Project: Predict Stock Prices Using Random Forest Regression Model in Python [Blog post]. [\[Link\]](#)
31. Serafeim, L. (2020). LSTM Time Series Forecasting: Predicting Stock Prices Using an LSTM Model. Towards Data Science. [\[Link\]](#)
32. Longmore, K. (2020). A Vector Autoregression Trading Model. Robot Wealth. [\[Link\]](#)
33. Dat Tan Trinh. (2022). Stock Price Forecasting Based on XGBoost and LSTM. [\[Link\]](#)
34. Rahman, M. A. (2023). Forecasting Stock Prices on the Basis of Technical Analysis in the Industrial Sectors of the UK Stock Market. *International Journal of Advances in Business and Management (IJABM)*, 2(1), 11-32. [\[Link\]](#)
35. Yahoo Finance. (2023). S&P 500 historical data 1990-2023. [\[Link\]](#)
36. Stanislaw Weglarczyk (2018). Kernel Density Estimation and Its Application. [\[Link\]](#)
37. Wooldridge, J. M. (2019). Introductory Econometrics: A Modern Approach. Cengage Learning. [\[Google Scholar\]](#)
38. Yukai, Y. (2022). A Comparative Study of the KPSS and ADF Tests in terms of Size and Power. [\[Link\]](#)

39. Zaiontz, Ch. (2018). Augmented Dickey-Fuller Test. [\[Link\]](#)
40. Prabhakaran,S. (2019). Augmented Dickey-Fuller Test. [\[Link\]](#)
41. Naushad, R. (2020). Interpreting Results of Dickey-Fuller Test for Time Series Analysis. Medium. [\[Link\]](#)
42. Chumachenko D. I. (2020). Mathematical models and methods for predicting epidemic processes: monograph. Kharkiv: TOV “Planeta-Print”, 180 pp.
43. Hyndman, R. J. (2018). Forecasting: Principles and Practice - Identifying the numbers of AR or MA terms in an ARIMA model. [\[Link\]](#)
44. Smigel, L. (2020.). How to Interpret ARIMA Model Results. [\[Link\]](#)
46. Nochovny, O.O. (2020). Information system for forecasting stock quotes using machine learning methods. [\[Link\]](#)
47. Alexander Dyakonov. (2022). Small data analysis - Quality Metrics. [\[Link\]](#)
48. Nicolas Vandeput. (2019). Forecast KPI - RMSE, MAE, MAPE, Bias. Towards Data Science. [\[Link\]](#)