Predictive methods in time series and machine learning of macroeconomic data

Master's Thesis

Máster en Modelización e Investigación Matemática, Estadística y Computación

UNIVERSITY OF ZARAGOZA

September 2023

Author:

Raúl Almuzara Diarte

Supervisors:

Juan Luis Fernández Martínez
Lucas Fernández Brillet

Resumen

Este trabajo trata sobre la modelización estadística de series temporales de precios para realizar predicciones. En primer lugar, se presenta un apartado de preliminares que incluye un resumen de aspectos generales de las series temporales, elementos económicos fundamentales del mercado de valores y particularidades de las series temporales financieras. Seguidamente, se presenta la metodología que seguiremos. Después, se muestra la serie principal de precios diarios de cierre de una empresa y otras covariables adicionales que reflejan información económica externa para intentar incrementar el poder predictivo. Tras ello, se describe el funcionamiento y la implementación de los modelos de Holt-Winters, Prophet y SARIMA en Python. El objetivo es desarrollar implementaciones que permitan reentrenar los modelos día tras día para realizar predicciones sucesivas del día siguiente aprovechando la información más reciente en cada momento. De esta forma, podemos simular los resultados que se obtendrían en un entorno real de inversión a lo largo de un año y desarrollar estrategias de predicción para el futuro. Se exploran diferentes configuraciones de modelos para optimizarlos y se evalúan los resultados con las pertinentes métricas. Finalmente, se describen las principales vías de investigación futura y se discute la significación de los resultados.

Abstract

This thesis deals with statistical modeling of price time series for making forecasts. First, a section of preliminaries is presented including a summary of general aspects of time series, fundamental economic elements of the stock market and particularities of financial time series. Next, we present the methodology that will be followed. Then, we show the main series of daily closing prices of a company and additional covariates reflecting external economic information to try to increase the predictive power. After that, we describe the operation and implementation of the Holt-Winters, Prophet and SARIMA models in Python. The objective is to develop implementations that allow retraining the models day after day to make rolling forecasts of the next day taking advantage of the most recent information at each moment. In this way, we can simulate the results that would be obtained in a real investment environment over the course of a year and develop prediction strategies for the future. We explore different model configurations to optimize them and evaluate the results with the relevant metrics. Finally, the main directions for future research are described and the significance of the results is discussed.

Contents

1 Introduction and motivation

In recent years, interest in data analysis and predictive methods has grown significantly. Data science, in its broadest sense, has acquired a fundamental relevance in decision making. Today, larger and larger amounts of data are becoming available, as well as more widespread access to the necessary computational power to analyze them. A proper analysis of the data using the relevant statistical and computer techniques makes it possible to extract valuable information. The use of the right tools can have a substantial impact on the profitability or efficiency of a company's activities. Especially in recent years, companies have become aware of the potential of data as a source of information to optimize processes or make predictions. Relatively accurate predictions can be very valuable for a company to anticipate possible risks or to choose a specific business strategy.

This project deals with the statistical processing of financial data from the stock market. The operation of the stock market is very complex, but it plays a fundamental role in the development of companies and, by extension, in the development of the economy. Depending on multiple economic and social factors, a company's shares have a price that fluctuates continuously according to the perceived value of the company. Even if one might think that stock price fluctuations and other macroeconomic variables are random and unpredictable, the appropriate statistical, deep learning or pattern recognition techniques may allow to make valuable forecasts for investors depending on changes in the market and predict risks. Statistical methods and artificial intelligence methods may be used for algorithmic trading and a thorough technical analysis of the market helps to get a deep understanding of the market behavior in the past and in the future. The growing demand for quantitative analysts specializing in financial mathematics reflects the importance of mathematical models in finance and the need for experts in predictive methods that can help make important economic decisions. As these decisions are increasingly data-driven, it is necessary to build quantitative models that learn from our information sets and provide value, avoiding investments based on emotion.

In this thesis, some important techniques of time series analysis will be shown with a special emphasis on the analysis of financial time series and the predictions that can be made on them. The theory of time series analysis is very rich from the statistical point of view and multiple statistical models have been developed, ranging from very intuitive ideas of regression to much more complex models involving modern artificial intelligence techniques. We will examine different methodologies used in the analysis of financial time series and explore some models and techniques, from the most classical ones of the last century to more innovative ones. Among others, knowledge of advanced statistics, time series, data mining and machine learning has been useful to carry out this work and is expected for the reader to a certain extent, although this thesis is intended to be self-contained. A nice reference which serves as an introduction to econometrics is [20]. Regarding the topic of time series, a good introduction can be found in [2]. Specifically as bibliography for financial time series, reference [22] is highly recommended and one of the most popular books in this domain.

This work has been carried out in collaboration with the Spanish start-up StockFink. It is a company specialized in the application of artificial intelligence techniques for stock market

analysis in order to make accurate and unbiased predictions, minimizing risk and maximizing returns. The models will be implemented in Python and they will be easy-to-use and customizable. This is also a good opportunity to learn and reflect in a self-contained way on some of the features of statistical models and what satisfactory and unsatisfactory results can be achieved when applied to this problem. Forecasting financial time series is undoubtedly a great challenge due to the irregularity they often manifest, but in any case, they constitute a good ground on which to carry out a project on data analysis and predictive techniques. Although stock market time series are famously difficult to forecast and there is no publicly available method to obtain guaranteed accurate results, we will compare the adaptability of each model when retrained day after day and see the most relevant results that can be achieved, as well as their significance.

In this project, we will focus on the prediction of the closing price of a stock on the following day, although the algorithms are designed with generality to predict as many days ahead at a time as necessary. We will begin with some preliminaries which are fundamental to understand the handling of time series and the economic context. This includes a description of time series basics (mathematical definition, components, statistical properties...), a summary of economics, companies and stock markets (relevance of stocks, operation of stock exchanges, share prices...) and specific information about financial time series (challenges, economic indicators...). Then, we will show the methodology that we will follow. Before entering into complex mathematical models, we will perform a primary exploratory analysis of the financial time series involved in this work to understand the nature of the data that will be modeled. After that, we will be ready to delve into the design of statistical time series models. The models analyzed are Holt-Winters, Prophet and SARIMA. For each one, a theoretical summary of its way of functioning will be presented, the particularities of its implementation will be explained and the results will be discussed along with the pertinent evaluation of the model's performance and relevant metrics. Finally, in the conclusion, we will reflect on the results obtained, propose possible specific directions for future work and make some valuable general comments on the modeling of financial markets from a mathematical point of view and the application of this type of techniques in this domain.

This document presents the analysis of a specific time series at a specific interval in order to show some specific results and graphs. However, the methods developed can be applied to any financial time series of the stock market and the configuration of the models can be freely modified, since the code developed is intended to automate the process and make the analysis generalizable. All datasets used in this work are public. The main data are the daily prices, which are provided by Yahoo! Finance and extracted from their APIs using [26]. The volume of shares is also obtained from this source. Regarding the development of the models in Python, the proposed rolling forecast algorithms are based on implementations which are documented in the corresponding websites for the libraries. Namely, you can find the documentation for the Holt-Winters method in [28], for the Prophet tool in [29] and for the SARIMA model in [30].

2 Preliminaries

In this thesis, we will mainly work with time series data, but first we need a good understanding of how to mathematically define a series of observations over time and their treatment in the field of statistics. We will then look at some basic concepts of finance and analyze some of the particularities of financial time series including those of stock market prices.

2.1 Time series basics

Let us begin with the most fundamental definition. A *time series* is a sequence of data points y_t, each with an associated time index t. More formally, an observed time series is one realization of a stochastic process composed of a set of random variables $\{Y_t\}$ indexed by integers representing days, weeks, months, quarters, years... In general, y_t is a vector in $\mathbb{R}^n$, where n is the number of observed time dependent variables at each given time and t is a label indicating the time step associated with each data point. If $n = 1$, the time series is called *univariate*. If $n > 1$, it is called *multivariate*.
We wish to create models that adequately represent the data of the past and with which we can make valuable predictions. The analysis and modeling of time series can sometimes be regarded as the application of regression techniques. Indeed, the two concepts may be interchangeable in some contexts, but it is important to highlight some general differences that make time series require a somewhat special treatment:

- The time dependence of the data in a time series establishes an intrinsic order in the structure of the data. The location of each piece of data with its corresponding time label is important. We pay attention to patterns over time and concepts such as *autocorrelation*, *stationarity* or *seasonality* appear. In a standard regression problem, data do not usually need to be in a particular order and the model does not have to distinguish between the position of each instance, unless specified.

- In general, the interest of time series analysis lies in the forecast of *out-of-sample* future values (*extrapolation*). It does not really make sense to design models to forecast values in times already past, although *in-sample* predictions can be evaluated to ensure that past data have been well modeled. In a regression problem, it may be interesting to make predictions directly within the range used for training (*interpolation*).

- As a consequence of the time structure, many time series with information of interest will be a realization of random variables that are not independent of each other.

Typically, a time series can be decomposed into several components:

- **Trend T_t:** The general long-term pattern of the time series.

- **Seasonal S_t:** Regular fluctuations of fixed period based on the season.

- **Irregular I_t:** The remainder after the other components have been removed from the original series. An unpredictable random error component.

Certain models may include additional components like cyclic components to account for rises and falls with no fixed period or holiday components to account for special events of the year. Putting all components together, we recover the original series. However, there is more than one way to do the decomposition.

- Adding the components defines an *additive* model:

$$y_t = T_t + S_t + I_t \tag{1}$$

- Multiplying the components defines a *multiplicative* model:

$$y_t = T_t \cdot S_t \cdot I_t \tag{2}$$

  which can be turned into a sum of components by taking logarithms:

$$\log(y_t) = \log(T_t) + \log(S_t) + \log(I_t) \tag{3}$$

Sometimes, the additive or multiplicative nature of the components themselves (trend and seasonality) is also discussed. An additive trend is usually related to linear trends, and an additive seasonality is usually related to series with oscillations having roughly the same amplitude over seasonal cycles. On the other hand, a multiplicative trend is usually related to non-linear trends, and a multiplicative seasonality is related to seasonal components with variable amplitude over seasonal cycles.

In time series analysis, the concept of certain properties varying or being constant over time is relevant. A time series is said to be *stationary* if its properties do not depend on the time at which the series is observed. Strictly speaking, there are two characterizations of stationarity. Let $\{Y_t\}$ be the associated random variables of a time series with realizations $\{y_t\}$ and $(t_1, \ldots, t_k)$ is a collection of k positive integers. We say that the time series $\{Y_t\}$ is *strongly stationary* if the joint distribution of $(Y_{t_1}, \ldots, Y_{t_k})$ is invariant under time shift. That is, the joint distribution of $(Y_{t_1}, \ldots, Y_{t_k})$ is the same as the joint distribution of $(Y_{t_1+s}, \ldots, Y_{t_k+s})$ for any integer s. Nevertheless, this condition is difficult to verify in practice. For this reason, a more relaxed definition is proposed. We say that the time series $\{Y_t\}$ is *weakly stationary* if $\mathrm{E}(Y_t)$ (expected value of Y_t) is constant and $\mathrm{Cov}(Y_t, Y_{t-l})$ (covariance between Y_t and Y_{t-l}) only depends on l, an arbitrary integer. Essentially, this means that the values of a weakly stationary time series fluctuate with constant variance around a fixed level. These conditions for weakly stationarity are easier to check in practical observations of time series and their approximate fulfillment is usually sufficient for practical applications. Therefore, when we refer to a series simply as *stationary*, we usually speak of weak stationarity.
Conversely, a *non-stationary* time series is one that does not satisfy some of the conditions above, i.e., its statistical properties change over time and they depend on the time at which the series is observed. Thus, a time series with a clearly defined trend component will likely be non-stationary since the mean would vary over time. A typical way to convert a non-stationary series into a stationary one is by applying differences. Differencing a time series consists of

applying the difference operator one or more times. The lag-l difference of order 1 of a time series y_t is

$$\nabla_{(l)} y_t = y_t - y_{t-l} \tag{4}$$

We can keep differencing for higher orders, although it will normally be sufficient to consider low orders. Special attention should also be paid to the fact that some data points are lost after differencing. For time series with trend component, we can expect to produce a detrended time series after applying the operator with $l = 1$ a few times. For time series with seasonal component, we can expect to produce a deseasonalized time series after applying the operator a few times with a lag equal to the period of the seasonal cycles. In case of having both types of components, we should start by eliminating the seasonal component through seasonal differences and then eliminate the trend through ordinary differences.

The correlation between two random variables is an important concept in probability and statistics as a measure of the strength of the linear dependence between them. In the study of time series as stochastic processes, a series is considered as a set of random variables and each observed value is considered as a realization of each random variable. Given the dependence that often exists between the different observations that make up a time series, this concept appears in the study of the *AutoCorrelation Function* (ACF). The ACF is a measure of the dependence between a time series y_t and its past (lagged) values. Under the weak stationarity assumption, the following expression describes the autocorrelation ρ_l between Y_t and Y_{t-l} for some lag l.

$$\rho_l = \frac{\mathrm{Cov}(Y_t, Y_{t-l})}{\sqrt{\mathrm{Var}(Y_t)\mathrm{Var}(Y_{t-l})}} = \frac{\mathrm{Cov}(Y_t, Y_{t-l})}{\mathrm{Var}(Y_t)} = \frac{\gamma_l}{\gamma_0} \tag{5}$$

A related concept is the *Partial AutoCorrelation Function* (PACF). For each lag l, the PACF is a measure of the correlation between Y_t and Y_{t-l} after eliminating the linear dependence due to the intermediate values.

A popular measure for model selection is the Akaike Information Criterion (AIC) defined as

$$\mathrm{AIC} = 2k - 2\log(\hat{\mathcal{L}}) \tag{6}$$

where k is the number of estimated parameters in the model and $\hat{\mathcal{L}}$ is the maximized value of the likelihood function for the model. Among the possible candidate models from a class of models for a specific train set, it is convenient to choose the one that minimizes the AIC measure in order to minimize the loss of information. In this way, we maintain a balance between the simplicity of the model (to avoid *overfitting* due to an excessive number of parameters) and the goodness of the model (to avoid *underfitting* due to a model without sufficient explanatory capacity).
After having obtained a model that we judge adequate, we can evaluate its generalization capacity in the interval where we reserved the test data. The goodness of the model can be evaluated through some metric of quantification of the deviation between the forecasts and the actual test values. A classically used metric for time series is the mean squared error, but for our specific

problem, we will also will be interested in the number of correct guesses as to whether the data is increasing or decreasing with respect to the previous data point.

2.2 Economics, companies and stock markets

Economics is a field that is no exception when it comes to the importance of data analysis. Especially in a field that deals with the management of issues directly related to money, there is an enormous interest in knowing trends and patterns well.

Companies are organizations that provide goods and services and play an important role in the development of a country's economy. However, the financing process is very complex and the concept of stocks comes into play. *Stocks* are securities that represent partial ownership of publicly listed companies. Companies issue stock as a method of raising money by attracting investors. *Shares* represent a unit of ownership of a stock. A shareholder has an ownership interest in a company and will hope to make a profit out of the investment when a stock rises in price or in the form of dividend payments. Sometimes, they can also obtain voting rights on certain corporate actions.

The prices of stocks, currencies, raw materials and other goods with value are constantly fluctuating. For example, the value of Microsoft shares, the cost of buying a bitcoin or the price of gold change all the time. In short, the goal of any investor is to *buy low and sell high*. In the case of companies, the holding time of a stock can be very short as in the case of intraday trading or very long in the case of expecting long term profits for some promising asset or industry. Also, investors can focus on a particular industry if they believe it is a key sector to maximize their returns or diversify their portfolio in order to reduce risk. In any case, it is necessary to study methods for making reliable forecasts in order to make smart investments.

As of 2023, in terms of market capitalization, the two largest stock exchange operators in the world are the *New York Stock Exchange* (NYSE) and the *National Association of Securities Dealers Automated Quotations* (NASDAQ). Both operate in the United States. At the end of 2022, the market capitalization of NYSE companies totaled around 22.7 trillion dollars and the market capitalization of NASDAQ companies amounted to around 16.2 trillion dollars. Both NYSE and NASDAQ are open for trading from Monday through Friday 9:30 a.m. to 4:00 p.m. Eastern Time, except during specific holidays. Roughly, a week has 5 trading days, a month has around 21 trading days and a year has around 252 trading days.

In order to evaluate the state of the market through various companies, we use *stock market indices*. A stock market index is a measurement of the market performance over time using the information of a representative section of the stock market. Some of the major American indices are the *S&P 500* (500 of the largest companies on stock exchanges in the United States), the *Dow Jones Industrial Average* (30 prominent companies listed on American stock exchanges) and the *NASDAQ Composite* (almost all stocks listed on the NASDAQ exchange). They can be seen as a measure of the overall situation of the economy and indicators of the general stock price levels for each period.

When we look at the information of a stock exchange on a certain stock in a certain trading period, we distinguish different figures:

- **Opening price:** The price at which a stock started trading at the beginning of a period.
- **High price:** The highest price at which a stock traded during a period.
- **Low price:** The lowest price at which a stock traded during a period.
- **Closing price:** The price of a stock at the end of a period.
- **Volume:** The number of shares traded during a period.

We may work with the Open, High, Low and Close prices for a given period and granularity as separate time series. However, they are commonly summarized and plotted in an *OHLC chart* or in a *candlestick chart*. In the OHLC chart, each period has a vertical line and two short horizontal lines. The vertical line is the range of prices defined by the low price and the high price. The horizontal line on the left marks the opening price and the horizontal line on the right marks the closing price. In the candlestick chart, each period has a *candle* with a real body representing the interval between the opening price and the closing price, an upper wick marking the high price at the top and a lower wick showing the low price at the bottom. In both charts, the color of a period is green if the closing price has been higher than the opening price and it is red if the closing price has been lower than the opening price. Both chart styles provide the same information and it is a matter of clarity or personal preference.

Figure 1: OHLC chart for the daily AAPL prices in the last quarter of 2022.

Figure 2: Candlestick chart for the daily AAPL prices in the last quarter of 2022.

Among investors, the term *bullish* is used to describe a market with rising prices and an optimistic sentiment with a positive attitude towards investment. In contrast, the term *bearish* is related to a market with falling prices and a pessimistic sentiment and reluctance to acquire new shares.

2.3 Financial time series

A *financial time series* is a sequence of data on prices, sales, indicators or any other relevant magnitude in an economic context over time. Financial time series analysis can sometimes pose a number of challenges:

- It can be difficult to extract clear long-term trends given the irregularity of the market and the constant changes in all directions that can be found in most stocks. The behavior of prices and economic indicators is usually irregular and nonlinear, so prediction is a complicated task. We can build models intelligently, but creating a general framework for making absolutely accurate predictions remains an open problem.

- Volatility is a problem when trying to make accurate predictions because high data variance will generate greater uncertainty. Heteroscedasticity (heterogeneity of variance over time) is also commonly present in clusters requiring an additional level of modeling. In certain times of economic crisis, volatility is more noticeable and makes it difficult to make safe investments and to understand the direction in which prices will move in the next time steps.

- Some seasonal patterns can sometimes be observed due to the nature of certain industries or areas of operation. However, in general, it is often difficult to extract a clear seasonal signal that repeats cyclically over time.

- Knowing past data may be useful for predicting future values, but it is not always sufficient. Market fluctuations are due to many complex factors. Some can be modeled by knowing real-time information about a company's activities and external economic indicators, then feeding that information into the model. However, for long-term investment plans there is always the risk of unexpected events. It is common to constantly retrain models by introducing updated data so that forecasts are more accurate and models keep up to date with changes in the market. It would be very easy to guess the behavior of the markets if all companies maintained an easily predictable trend over time. Unfortunately, a long period of growth may not continue for all future time. Likewise, a stock that has been depreciating for a long time may suddenly start to rebound. There is always a risk of unpredictable events occurring, but in any case, a good model must be able to adapt to new price changes that are introduced over time, whatever their nature.

- The information of interest is often presented in the form of series with different time granularity and it can be difficult to consider all the information together. Moreover, for a particular variable, there may be changes in the trend depending on the time frame (e.g. bullish behavior over weeks, but bearish over months).

Among the types of prices defined in the preceding subsection, a variable that is commonly forecasted is the closing price, although the other types of prices can also play an important role in forecasting models. In general, let P_t the price of an asset at time t. The ultimate goal is to forecast the future values of the data series P_t. Nevertheless, it is also possible to formulate a description in terms of *returns*. The one-period *simple return* R_t gives an idea of the relative change of a price from time $t-1$ to time t:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{7}$$

Due to convenient statistical properties, it is also common to speak of *log returns* r_t:

$$r_t = \log(1 + R_t) = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(P_t) - \log(P_{t-1}) \tag{8}$$

whose usefulness can be linked to the convenience of taking logarithms and differencing a time series. Note that, if the simple returns are not very large in absolute value, the Taylor expansion of the logarithm allows the reasonable approximation $r_t = \log(1 + R_t) \approx R_t$.

We will focus on financial time series of a specific company. The price time series are extracted from Yahoo!'s publicly available APIs and loaded into Python through the open-source library `yfinance`. This allows direct and automatic loading of the data at the desired date and granularity for any listed company with an associated *ticker*. Unless there is an error in Yahoo!'s financial data reporting service, there should be no missing data, except for the days and times when it is already known that the stock exchanges do not operate.

Historically, there is a large number of statistical prediction models that have been proposed to be applied to the problem of market forecasting, with varying degrees of success. When building a model, success may depend primarily on two types of aspects.

- The kind of model used (autoregressive models, neural networks, etc.) and the tuning (hyperparameters and additional configuration options).

  In this thesis, a variety of models will be explored and described in a self-contained manner and their use will be justified in the following sections.

- Features on which the prediction is based (economic and social indicators or additional exogenous variables which are expected to have a significant influence).

  There are hundreds of potential information sets that influence the amount we want to forecast. The dynamics of the market and of each company are very complex and identifying the most representative factors is a difficult task. We may classify this kind of covariates into two groups:

  - General macroeconomic indicators: Global magnitudes on the state of the economy. For example, the gross domestic product or stock market indices like the S&P 500, DJIA and NASDAQ Composite.
  - Company-specific variables: Data about a company's performance, interactions and perception. For example, the volume of shares traded in a period, the earnings that American companies report every quarter, a measure of interest in the company over time (e.g. Google Trends).

In complex econometric time series, we may try to forecast one series with the help of other economic variables of potential interest. However, the specific correlations might not be clear. After all, the economy is a complex ecosystem where we can measure some quantities, but not necessarily fully understand how they interact at any given time. Also, it is clear that investors react to the market to base their decisions, but the market also reacts to investors. Is a specific time series useful in forecasting another time series? This is the statistical concept of *Granger causality* [19]. We say that a time series x_t *Granger-causes* a time series y_t if the past values of x_t contain statistically significant information that helps forecast y_t. One of the main challenges of enhancing models is to select a set of covariates that provide valuable information about external circumstances that are expected to help explain the price dynamics of the main series. If possible, it is desirable that all the time series that will be considered simultaneously have the same granularity. This is not always possible. This is especially problematic when dealing with high frequency data where a lot of data is collected at a high speed, but the other additional information has a much lower resolution. However, there is no need to resort to the example of high-frequency trading because we may also encounter this problem with certain macroeconomic indicators that are not updated with a high frequency (such as the GDP or a company's quarterly earnings). Also, if necessary, covariates should be converted to a scale that is compatible with the main data series to avoid numerical errors and biases towards certain variables if there are notable differences in the orders of magnitude.
It is essential to have a good understanding of the exact information available at any given time for a realistic analysis. That is, without cheating by taking advantage of future data that we would not have available in the present if it were a real forecasting scenario. *Data leakage* would misrepresent the true capacity of the models by possibly giving better results than they should.

3 Methodology

As is customary in the process of building machine learning models, the first step is to divide the dataset into subsets: a *training* set to fit the model and a *test* set to evaluate its performance with data it has not seen before. From the training data, and within the limitations of the model complexity, the model learns the most appropriate parameter values. If the model has sufficient capacity, it is expected that the training set data will be well represented and that the model can generalize well to obtain good results on the test subset. In a standard machine learning problem with independent instances, the train-test split could be done as a simple random partitioning of the dataset. However, in a time series with information of interest to be modeled, the observations will not be independent because of an underlying trend or seasonal behavior that makes the observation of each time dependent on the past values. In that case, the partition into subsets must be done more carefully since we have to respect the time order (it would make no sense to use data from the future to predict values from the past). It is obvious that models should be trained on earlier data and tested on later data since that is the way in which time is passing. Thereafter, the cross-validation process practiced on time series is based on the training of successive windows over time across the entire test set. Not only is it a technique that allows us to demonstrate well the potential of each model over time, but for our particular problem it is the most realistic way to simulate their performance in a real investment environment.

There are several ways to carry out this procedure. Here, we will use the technique of forecasting on a *rolling basis*. We establish from the beginning a set of available data, set an initial position of separation between training and test data (this is just an initial partition) and define a time window size. With the training data assigned at each iteration, a model is trained and prepared to predict in as many units of future time as desired. Once the forecast has been made, the training window is moved into the future and we continue training and predicting over the desired test interval. If the computational power allows it, when using these models to make real predictions, it is essential to constantly update them with the latest information, since the market is very changeable and very long-term predictions with old data are likely to give poor results. Also, it seems more convenient to set a specific window size rather than to keep on expanding it by retaining very old data, since stock price series have a limited memory.

In summary, we first define an initial cut-off point, which is the end of 2021 in our case. The first model is trained with the last data points of 2021 to make a forecast for the first trading day of 2022. The training window is moved one day into the future incorporating this last data point of 2022 and removing the oldest data point of 2021 (thus keeping the same training window size). The model is retrained and we keep retraining and forecasting the next day for the whole 2022 (251 trading days). In principle, this could be considered a sufficient amount of data points to evaluate the actual capacity of the models. It would be ludicrous to evaluate the accuracy of the models in a very small interval, but it could be justified in some cases where a stock has a memory much more limited than a year. Besides, a small test set could lead to overly optimistic or pessimistic results that do not reflect the actual performance of the models. However, a large test set also involves considerable computation time since we have to train 251 models for each specific combination of the hyperparameter space.

Two metrics have been programmed: the root mean squared error and the proportion of correct predictions as to whether the prediction of the rise or fall of the stock (compared to the previous day) coincides with reality. Recall that the root mean squared error is

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{9}$$

where n is the number of data points ($n = 251$ in our case since there are 251 trading days in 2022 where we will test the predictions), y_i are the actual data points and $\hat{y}_i$ are the forecasts. As for the number of rises and falls, we simply compare whether the current day's price is higher or lower than the previous day's price. For ease of implementation, although the test set consists of the 251 trading days that make up the year 2022, only 250 data points of rises and falls are recorded, since we have decided to make the calculations exclusively on this set and the first data of the test set is not compared to the last data in the training set (also, 250 is a nicer number that makes it easier to show exact proportions without rounding).
Both the mean squared error metric and the metric for correct predictions in rises and falls are very relevant to evaluate the performance of a sequence of models. However, they are completely independent and the same sequence of models can produce good results in one metric and bad results in the other simultaneously. As an idea of model selection, it is desirable to choose a model that provides an acceptable result for both metrics, maintaining an adequate balance when possible. Each type of model will be trained testing multiple combinations of hyperparameters to evaluate the performance. Consecutive models are trained moving the training window day after day. This preliminary tuning process will be first done without covariates and the best hyperparameters will be chosen. The specific hyperparameters depend on the class of model and will be described in the corresponding sections. In the case of the SARIMA model, after having this optimal selection, we will train the corresponding 251 models, now adding the exogenous variables. The general methodology would consist of carrying out a similar analysis for each stock of interest in the days prior to the one we want to predict to discover the best configuration, but here we will use Apple's prices to show specific results.
Considering that the main series to be predicted is the daily closing price series of a company, after a bit of exploration, some covariates that seemed to add interesting predictive value and do not overload the models excessively were:

- The daily opening price series of the corresponding company (Apple in this analysis), closely related to the associated closing price.
- The daily opening price series of the S&P 500 index, reflecting the global state of the economy since it is computed considering the performance of 500 of the largest companies listed on stock exchanges in the United States. It is important since company share prices may not necessarily be independent of each other.
- The daily volume of Apple shares traded, showing the total amount of transactions (buys and sells).

Recall that the daily closing price is the price of the stock when the exchange closes at 4:00 p.m., while the daily opening price is the price of the stock when the exchange opens at 9:30 a.m. The

opening price can potentially help to focus the forecast on a more realistic range of values, as the opening price will normally not be too far from the closing price on a typical trading day, thus reducing the risk of an extreme deviation. By including the actual opening price data of the day for which we want to predict the closing price in the afternoon, we are implicitly assuming that the prediction is not made the day before, but the day itself in the morning as soon as the exchange opens. To make the forecasts of univariate series models, the value of the exogenous variables has to be provided for the same time of the forecast before making such forecast. Therefore, we will make use of this value of opening price which is available first thing in the morning instead of providing a guess. In any case, the opening price at which the stock market opens in the morning is close to the closing price at which it closed the previous day (except for some minor variation since companies continue their activity while the stock exchange is closed), so this extra information that we provide to make the forecast at the same time step is already approximately known. This applies to Apple's and S&P 500's opening prices. As for the volume variable, we have no choice but to provide an educated guess for the day we are going to forecast. We may assume that the volume of the following day will not be very different from the volume of the previous day, so a conservative guess is to copy the same volume from the previous day.

4 Exploratory data analysis

In order to display specific results and graphs, the main financial time series we will be working with will be Apple's daily closing price series. The ticker associated with Apple in the stock market is AAPL and prices are in dollars. The code and the methodology of analysis are generalizable to any company that we want to analyze in any time interval. To set a specific time interval, we will load all the daily data available between 2015 and 2022. The idea is to make predictions for the year 2022 and we will be able to assess the accuracy of the models since we know the actual prices that were observed throughout that year. These models will be trained with data prior to 2022. Nevertheless, due to the rapidly changing dynamics of the stock market series, it will not always be useful to work with very old data and it will only be necessary to use data not much further from 2021.
Let us introduce the main time series of prices.

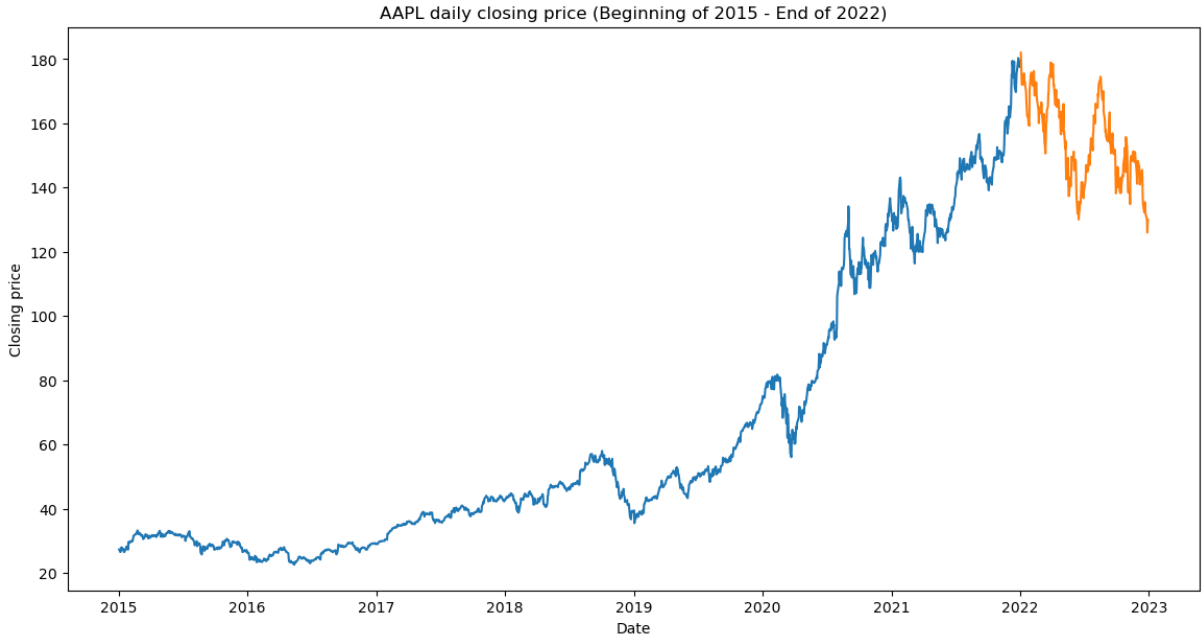

Figure 3: Apple's daily closing prices from the beginning of 2015 to the end of 2022. The blue data correspond to prices in 2015, 2016, 2017, 2018, 2019, 2020 and 2021. The orange data are the prices in 2022 and we will test the predictions in this period (251 trading days in 2022).

In general, we observe a fairly steady growth over the years, although it seems to have slowed down in the last year. Also, it seems that the oscillations become more pronounced as prices become larger generating more variance. In terms of notable events, there is no doubt that the COVID-19 crisis from March 2020 had a negative impact on prices, although they rebounded afterwards in this case. This event had a large effect on many financial series. Other periods of sustained growth or decline may also be due to the sustained impact of other specific events. More generally, another event that had a strong impact on price time series dynamics is the 2008 financial crisis (not shown in the above series period). More specifically for Apple, we can point to the global chip shortage of recent years, which is certainly not positive for the company's value.

Let us take a closer look at the components of the above time series. We will show two decompositions: one decomposition of the series in Figure 3 taking all data since 2015 and another decomposition of the series but taking only data since 2020, which is more related to more recent price dynamics.

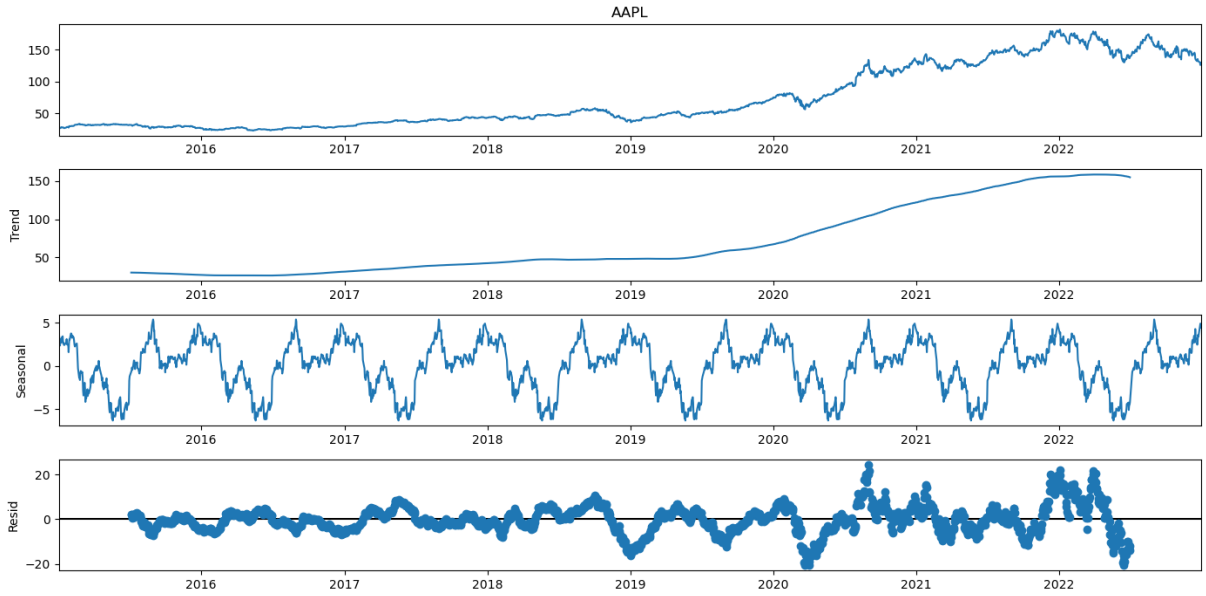

Figure 4: Additive seasonal decomposition (trend + seasonal + irregular component) of the Apple's daily closing price series since 2015.

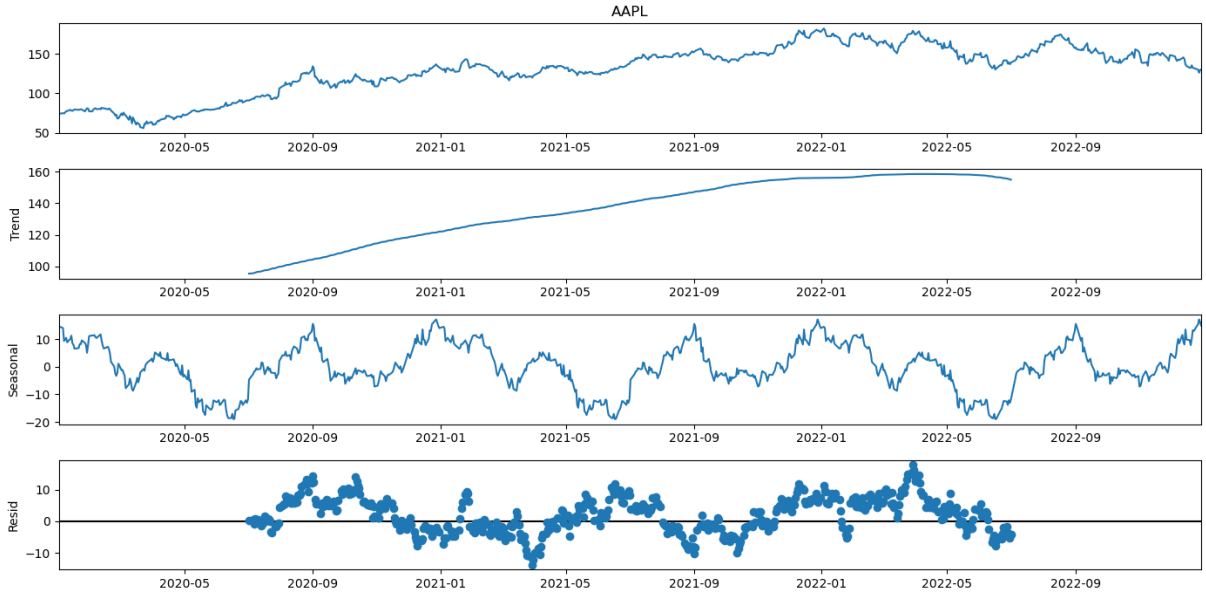

Figure 5: Additive seasonal decomposition (trend + seasonal + irregular component) of the Apple's daily closing price series since 2020.

The trend component shows a smooth version of the series where the variation in the speed of price growth in each year can be seen.

The seasonal component captures information about a possible periodic behavior, although

relatively weak. However, we can appreciate an interesting feature: the seasonal component peaks around early January and early September. On the one hand, the January spike could be explained by the company's possible increase in sales at the end of each year. It might also be attributable to the so-called January effect, a widespread hypothesis that there is a seasonal behavior in the financial market causing prices to increase in the month of January. On the other hand, more specific to Apple, there does appear to be an increase in Apple's popularity each September, which could translate into higher perceived value for the company (see, for example, the Google Trends measure in [27]). In this case, we have performed decompositions with period 252 (average number of trading days in a year) to analyze possible behaviors that repeat at specific times each year. However, when developing the models, we will study shorter periods as there may also be weekly or monthly patterns, with which it is more appropriate to work when considering short training windows.

So far, the daily closing price series has been shown. This is the main series we are interested in predicting. An exogenous variable that we can naturally consider is the opening price series.

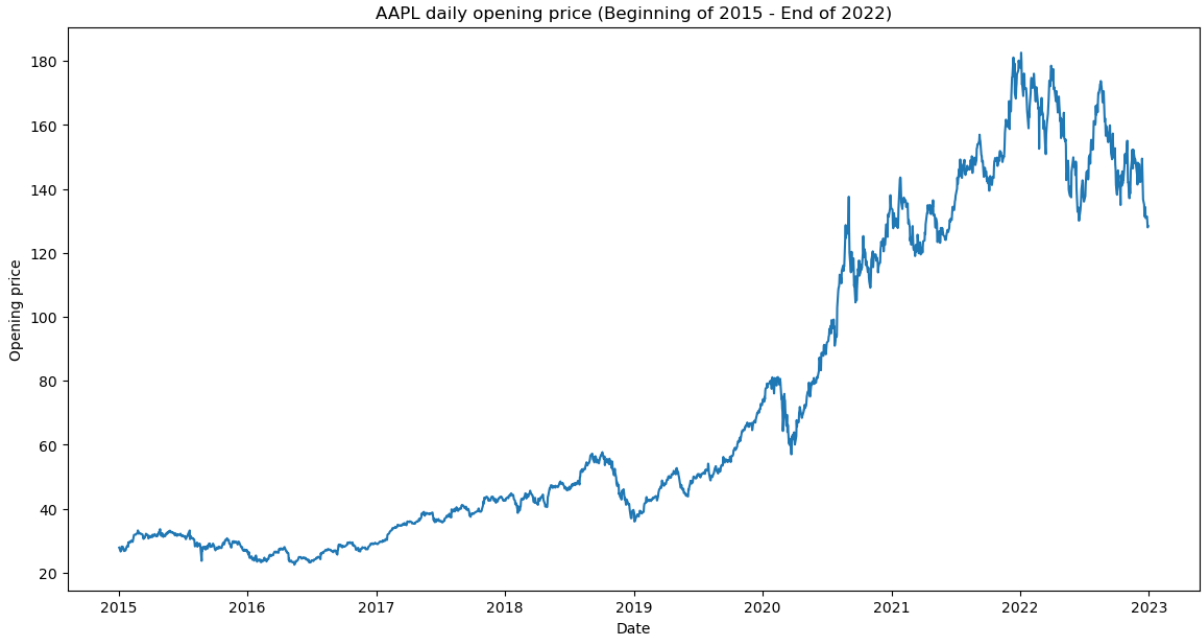

Figure 6: Apple's daily opening prices from the beginning of 2015 to the end of 2022.

The shape of the opening price series is evidently very similar to the shape of the closing price series in figure 3 since the truly remarkable variations within a time series are observed in the long term and not within a given day (at least for a stable stock without a large intraday variance).

One of the most commonly followed equity indices is the S&P 500 (we will use the ticker SPY). We will include it in the analysis, since the companies in a capitalist environment interact with each other and Apple's performance is dependent on the performance of other companies and the general state of the economy.

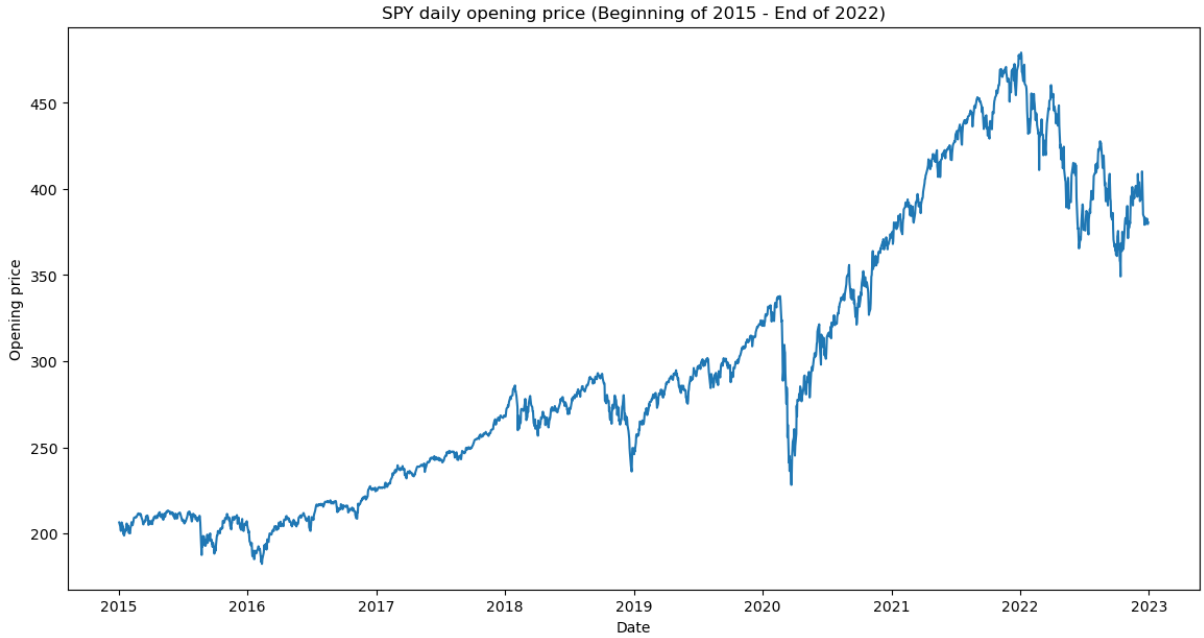

Figure 7: SPDR S&P 500 ETF Trust's daily opening price (2015-2022).

In a sense, it bears some resemblance to Apple's closing price series, but it also includes information from other companies.

Now, consider the daily volume, i.e., the number of shares that have changed hands each day.

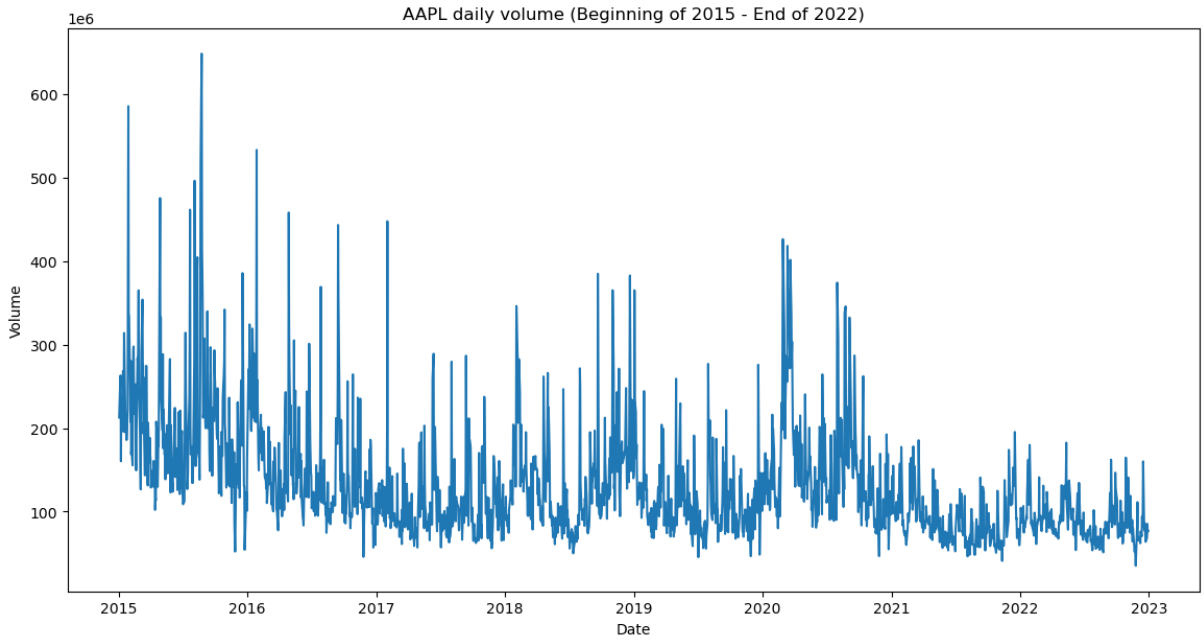

Figure 8: Apple's daily volume (in millions) (2015-2022).

Given the high order of magnitude of this variable, it will be necessary to transform it. Actually, the data we will be working with will be recent (from 2021 onward), where the order of volume remains relatively stable. By working with volume in millions, we have a variable in a similar order of magnitude to that of prices.

5 Exponential smoothing methods

As an introduction to the application of time series analysis methods, we will start with one of the most classical techniques: *exponential smoothing*. This is a method that has been classically applied to the study of financial time series because of its simplicity in summarizing past data and generating extrapolations. In addition, it illustrates well the process of exploiting the information of the components of a series as we add layers of complexity. They have few hyperparameters and can be implemented relatively quickly. The results can be moderately good depending on the version used and the components that are clearly present in the time series.
We will now present a summary of the underlying theory, although more extensive explanations can be found in [9] and [28].

5.1 Theory

The general idea of exponential smoothing methods is that past observations of the time series are assigned exponentially decreasing weights. In this way, we give more importance to recent observations in the series and less and less importance to observations from the increasingly distant past.
Depending on the class of exponential smoothing method, we have a different number of equations. In any case, we can write them in terms of a *measurement equation* for the dynamics of the observed variables (main series y_t) and some *state equations* for each of the series components that we want to consider (level l_t, trend b_t and seasonal s_t components). These can be arranged to obtain a proper formulation of a *state space model*. From now on, we will consider this framework that leads to the formulation of an ETS model (Error, Trend, Seasonal). If a trend and/or seasonal component are to be considered, there are various ways of writing the equations depending on the additive or multiplicative nature of the components. To write them in a general form, consider the following operations:

- $\oplus_b$: the forward operation associated to the trend (addition if trend is additive and multiplication if trend is multiplicative).
- $\ominus_b$: the inverse operation associated to the trend (subtraction if trend is additive or division if trend is multiplicative).
- $\odot_d$: the operation linking the trend and the damping parameter (multiplication if trend is additive, power if trend is multiplicative).
- $\oplus_s$: the forward operation associated to the seasonality (addition if seasonality is additive and multiplication if seasonality is multiplicative).
- $\ominus_s$: the inverse operation associated to the seasonality (subtraction if seasonality is additive or division if seasonality is multiplicative).

In order of increasing complexity, the three main exponential smoothing methods are:

- **Simple exponential smoothing:** The most basic version of the exponential smoothing method.

The model can be expressed in terms of 1 forecast equation and 1 smoothing equation for the level. If the data at time $t-1$ is known, the 1-step ahead forecast for time t is given by:

$$\hat{y}_{t|t-1} = l_{t-1} \tag{10}$$

$$l_t = \alpha y_t + (1-\alpha) l_{t-1} \tag{11}$$

The parameter is α (the smoothing parameter for the level). We specify the process together with an initial value for the level smoothing component.

An important limitation with this approach is that this method can only forecast the level component of the time series. The simplicity of the model makes the forecasts flat and equal to the last level component (before undoing any potential transformations that may have been performed on the data). In some cases, a flat forecast may be more accurate than a complex forecast in some very specific interval. Nevertheless, in general, a flat forecast is extremely simplistic and does not capture bullish or bearish trends or seasonal fluctuations at all. It may be sufficient in case we want to smooth the behavior of a series in past data, but it might not be the most adequate method as a predictive model.

- **Double exponential smoothing:** Also known as the Holt linear method. This time, we consider the effect of an underlying trend in the series.

  The model can be expressed in terms of 1 forecast equation and 2 smoothing equations (level and trend). Consider that the data at time $t-1$ is known and we are interested in the 1-step ahead forecast for time t. However, we must consider that the nature of the trend can be additive or multiplicative and different models can be designed. A general form is:

$$\hat{y}_{t|t-1} = l_{t-1} \oplus_b (b_{t-1} \odot_d \phi) \tag{12}$$

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} \oplus_b (b_{t-1} \odot_d \phi)) \tag{13}$$

$$b_t = \frac{\beta}{\alpha}(l_t \ominus_b l_{t-1}) + \left(1 - \frac{\beta}{\alpha}\right)(b_{t-1} \odot_d \phi) \tag{14}$$

  The parameters of the model are α (the smoothing parameter for the level), β (the smoothing parameter for the trend) and, optionally, ϕ (the damping parameter for the trend). We specify the process together with an initial value for the level smoothing component and an initial value for the trend smoothing component.

  For time series with no seasonal component, this model offers a simple framework for smoothing and predicting. Now, both the level and the trend can be taken into account when making forecasts. For many purposes, this may be enough, although we are going to explore a more complete model.

- **Triple exponential smoothing:** Also known as the Holt-Winters method and the one we will focus on.

The model can be expressed in terms of 1 forecast equation and 3 smoothing equations (level, trend and seasonality). Consider that the data at time $t-1$ is known and we are interested in the 1-step ahead forecast for time t. In this case, we consider that the nature of the trend can be additive or multiplicative and that the nature of the seasonality can be additive or multiplicative. A general form is:

$$\hat{y}_{t|t-1} = (l_{t-1} \oplus_b (b_{t-1} \odot_d \phi)) \oplus_s s_{t-m} \tag{15}$$

$$l_t = \alpha(y_t \ominus_s s_{t-m}) + (1-\alpha)(l_{t-1} \oplus_b (b_{t-1} \odot_d \phi)) \tag{16}$$

$$b_t = \frac{\beta}{\alpha}(l_t \ominus_b l_{t-1}) + \left(1 - \frac{\beta}{\alpha}\right)(b_{t-1} \odot_d \phi) \tag{17}$$

$$s_t = \gamma(y_t \ominus_s (l_{t-1} \oplus_b (b_{t-1} \odot_d \phi))) + (1-\gamma)s_{t-m} \tag{18}$$

The parameters of the model are α (the smoothing parameter for the level), β (the smoothing parameter for the trend), γ (the smoothing parameter for the seasonality) and, optionally, ϕ (the damping parameter for the trend). We specify the process together with an initial value for the level smoothing component, an initial value for the trend smoothing component and as many initial values as seasonal periods in a cycle (m) for the seasonal smoothing component.

This is the model we should use for a time series if we wish to take advantage of its trend and seasonality components. It will be necessary to check if the additive or multiplicative approach for each component is more appropriate. More complex forecasts may better reflect some oscillations that are not predicted by the other two methods.

In all the models above, the true values y_t can be updated considering additive errors:

$$y_t = \hat{y}_{t|t-1} + e_t \tag{19}$$

or considering multiplicative errors:

$$y_t = \hat{y}_{t|t-1} \cdot (1 + e_t) \tag{20}$$

The choice affects the prediction intervals.
In this case, we will try to optimize appropriate Holt-Winters models. When fitting a model, the parameters α, β, γ, ϕ and the initial states of the components are those that maximize the likelihood function for the model.

5.2 Implementation

The model calculations are performed using the routines defined in the `statsmodels` library and, in particular, using the `ETSModel` implementation ([28]). We will prepare an algorithm for obtaining rolling forecasts that will automate the retraining of the model day after day throughout the year 2022. As explained in the methodology section, we will finally end up with forecasts for the 251 trading days of 2022 through 251 models trained with the most recent data available before each day. Details on the specific implementation of the algorithm can be found

in the code in the appendix. However, we must prepare the conditions of the simulation well and the models require the specification of various hyperparameters.
We will study a sufficiently large number of hyperparameter combinations. For each one, the successive re-training process described above will be performed. For the Holt-Winters model, the following variables have been tuned testing the values indicated below:

- **Training window size:** A priori, it is not clear how many days prior to the one we want to make the prediction we should use. The changing dynamics of the market suggest that we should not use data older than one year since market conditions more than a year ago may be radically different from today. Therefore, we will consider 4 different training window sizes: 2, 3, 6 and 12 months. These are 42, 63, 126 and 252 trading days, respectively.
- **Seasonal period:** The Holt-Winters method requires the specification of a period associated with the seasonal part. Given the relatively small size of the training window size, we will study relatively short seasonal patterns. We will consider weekly and monthly cycles (period 5 and 21, respectively).
- **Trend type:** Additive or multiplicative.
- **Seasonality type:** Additive or multiplicative.
- **Days forecasted:** Although we could forecast the price for several days with each model, we will always perform the retraining process for each day, thus forecasting just the price of the next day with each model.

In total, these choices imply that we have to test $4 \times 2 \times 2 \times 2 \times 1 = 32$ combinations of values. For each one, 251 models are trained throughout 2022. For each of the 32 combinations of hyperparameters, the output is the list of predictions for each day of the year 2022 along with 95% confidence intervals (considering the models with additive errors), the series of residuals for each of the 251 models and the list of parameters for each model.

5.3 Results

We will quantify the goodness of fit of the models with the mean squared error and the proportion of correct predictions as to whether a rise or a fall has been correctly predicted (out of 250). Below, we show all the combinations of the values described in the implementation subsection and the corresponding metrics.

Window size	Seasonal period	Trend	Seasonality	RMSE	Correct predictions
42	5	Additive	Additive	3.7609	127 (50.8%)
42	5	Additive	Multiplicative	3.7377	128 (51.2%)
42	5	Multiplicative	Additive	4.2196	122 (48.8%)
42	5	Multiplicative	Multiplicative	3.8137	125 (50.0%)
42	21	Additive	Additive	5.0342	138 (55.2%)
42	21	Additive	Multiplicative	5.1144	141 (56.4%)
42	21	Multiplicative	Additive	5.2925	137 (54.8%)
42	21	Multiplicative	Multiplicative	5.2715	142 (56.8%)
63	5	Additive	Additive	3.6070	129 (51.6%)
63	5	Additive	Multiplicative	3.5997	124 (49.6%)
63	5	Multiplicative	Additive	3.8862	134 (53.6%)
63	5	Multiplicative	Multiplicative	3.5991	127 (50.8%)
63	21	Additive	Additive	4.2145	139 (55.6%)
63	21	Additive	Multiplicative	4.1897	137 (54.8%)
63	21	Multiplicative	Additive	4.9338	131 (52.4%)
63	21	Multiplicative	Multiplicative	4.1846	136 (54.4%)
126	5	Additive	Additive	3.4912	130 (52.0%)
126	5	Additive	Multiplicative	3.4905	129 (51.6%)
126	5	Multiplicative	Additive	3.4840	132 (52.8%)
126	5	Multiplicative	Multiplicative	3.4897	129 (51.6%)
126	21	Additive	Additive	3.7260	138 (55.2%)
126	21	Additive	Multiplicative	3.7155	139 (55.6%)
126	21	Multiplicative	Additive	3.7193	136 (54.4%)
126	21	Multiplicative	Multiplicative	3.7102	138 (55.2%)
252	5	Additive	Additive	3.4502	127 (50.8%)
252	5	Additive	Multiplicative	3.4530	126 (50.4%)
252	5	Multiplicative	Additive	3.4557	127 (50.8%)
252	5	Multiplicative	Multiplicative	3.4552	126 (50.4%)
252	21	Additive	Additive	3.5658	137 (54.8%)
252	21	Additive	Multiplicative	3.5679	136 (54.4%)
252	21	Multiplicative	Additive	3.5636	136 (54.4%)
252	21	Multiplicative	Multiplicative	3.5696	136 (54.4%)

Table 1: RMSE and number of correct predictions of rises and falls throughout the year 2022 for each combination of values.

Interestingly, most combinations give rise to proportions of correct predictions over 50%, although in many cases, the improvement may not be more significant than the expected outcome of a coin toss. However, there are some interesting patterns. On the one hand, the root mean squared error tends to be lower in the combinations where we consider seasonal cycles of length 5 days. On the other hand, the number of correct predictions of rises and falls tends to be higher in the combinations where we consider seasonal cycles of length 21 days. In view of these results, we try to maintain an appropriate balance by trying to minimize the first metric and maximize the second. In this case, a relatively optimal combination could be the one with a training window size of 126 days, seasonal cycles of 21 days, additive trend and multiplicative

seasonality. This combination provides a high number of correct predictions of rises and falls while having an acceptable RMSE. This combination of values gives rise to a sequence of Holt-Winters models that have worked best during 2022 and could be expected to work well in later dates for this particular series. For the rest of this section, we will focus on the results obtained with this choice.

Before thinking of any statistical model, the baseline that comes to mind is to consider that the best prediction that can be made is to assume that the next day's closing price is the same as the previous day's and that rises and falls can only be predicted correctly 50% of the time. This baseline has been found to provide an RMSE of 3.4147. The RMSE of the selected Holt-Winters model is a bit behind with its RMSE of 3.7155, but the acceptably good accuracy for this kind of problem of 55.6% is better than making naive random predictions.

For each day, each model that we train has its own set of parameters. The number depends on the seasonal period chosen. There are always a smoothing level parameter, a smoothing trend parameter, a smoothing seasonal parameter, a damping trend parameter, an initial value for the level and an initial value for the trend. However, there are 5 extra initial values for the seasonal part if the period considered is 5 days or 21 extra initial values for the seasonal part if the period considered is 21 days. We fit a model for each of the 251 trading days that make up the year, so we have 251 different sets of parameters associated with the day on which we intend to make the forecast. Taking into account all these fitted models, we present below the range in which the smoothing parameters and the damping trend oscillate depending on the day.

Parameter	**Minimum value**	**Maximum value**
Smoothing level	0.8429	0.9999
Smoothing trend	0.00009	0.13817
Smoothing seasonal	0.00000001	0.00001679
Damping trend	0.80	0.98

Table 2: Range of the smoothing level, smoothing trend, smoothing seasonal and damping trend parameters among the 251 models fitted throughout 2022 with 126 training days, seasonal cycles of 21 days, additive trend and multiplicative seasonality.

In view of the results, we will find many cases in which the modeling of the series is practically done through the modeling of its level, since the parameters associated with the trend and seasonality may be insignificant in many cases. This is due to the difficulty of capturing very clear and stable trends and seasonal patterns over time. However, in some days, the contribution of the trend and seasonal parts can play a relevant role.

We take a look at the predictions obtained for each day along with the actual values of the prices that occurred on those days.

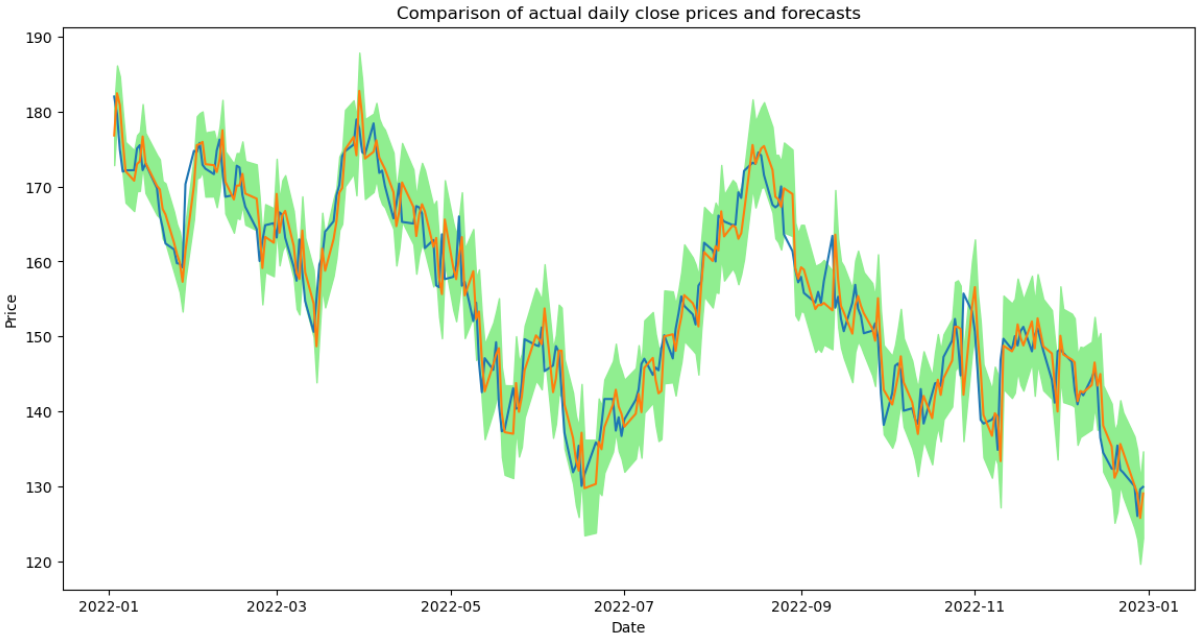

Figure 9: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the Holt-Winters method. In green, 95% confidence intervals.

It is essential to clarify how the forecasts have been constructed. The forecasts shown in the graph above have been obtained through rolling forecasts. A model is trained with the most recent data available up to that date and predicts the following day. The training window is moved one day forward, a new model is retrained and the next day is predicted. This is done successively over the course of a year to provide a sufficient amount of validation data. Thus, the above predictions are the result of 251 sequentially trained models (obviously, a single model would not be able to reproduce with such accuracy and without external information the price dynamics over a whole year). In principle, the predictions remain close to the actual values and the model is able to adapt dynamically in magnitude according to the new information that is introduced day after day.

Now, there is a finer way to consider the rises and falls in terms of the magnitude of the associated return with respect to the previous day. Having defined the return as the relative change in price with respect to the previous day, consider this quantity in percent for each day. There are some days where the price has not moved too much compared to the previous days, whereas there are other days with more drastic rises and falls. Thus, we will classify the days into three classes. Class 0 will correspond to the cases where the return with respect to the previous day is between -0.5% and 0.5% (lateralization or sideways movement). Class 1 will correspond to the cases where the return with respect to the previous day is lower than -0.5% (substantial fall). Class 2 will correspond to the cases where the return with respect to the previous day is greater than 0.5% (substantial rise). This classification is made for both the time series of forecasts and the time series of actual prices. Therefore, the results of the classification are summarized in a confusion matrix where we compare actual vs. predicted classes. In this case, this is the confusion matrix that has been obtained.

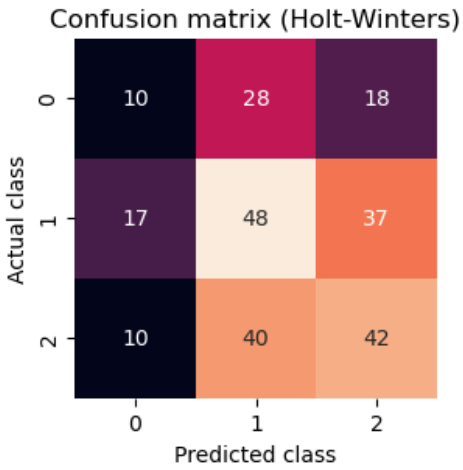

Figure 10: Confusion matrix of the Holt-Winters models comparing the classes of the actual daily closing prices and the classes of the forecasts. Class 0: Lateralizations (returns between -0.5% and 0.5%). Class 1: Returns lower than -0.5%. Class 2: Returns greater than 0.5%.

We see some deficiencies. The lateralizations are not really well predicted. As for substantial rises and falls, the numbers in the diagonal are the highest of their corresponding rows, but they do not represent more than 50% of these rows, so the predictions tend to deviate a bit. We will soon see if other statistical models perform a bit better in this aspect.

5.4 Additional comments

An important limitation of the classical Holt-Winters method is that it does not allow the natural inclusion of exogenous regressors. In recent years, some extensions have been proposed. However, forecastability issues may arise (see [10]). Research in recent years goes in the direction of including covariates in the framework of the state space formulation (see [15]). Another slightly more unsophisticated alternative could be to regress the main time series against the exogenous variables and then use the Holt-Winters method on the residuals from this original regression (outline in [12]). The proper addition of covariates will be discussed in more depth in the section for the SARIMA model.

6 Prophet models

Prophet is a modern forecasting procedure for time series. It was developed by Facebook and published as open-source software in 2017 (original publication in [21]). One of the model's greatest strengths is that the fitting process is extremely fast. In a matter of seconds, it is generally possible to estimate a model, obtain its components and make predictions. This is especially convenient taking into account that we wish to retrain the models many times and test many combinations of hyperparameters. In addition, it has functionalities to make the training and forecasting process as automatic as possible. However, it is possible to tune hyperparameters, incorporate covariates, consider special effects throughout the year and introduce domain knowledge in order to adapt the forecasts to the characteristics of a time series. It is based on an additive model capable of handling non-linear trends along with seasonality on different levels and distinctive holiday effects.

6.1 Theory

The time series is decomposed as an additive model:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon(t) \tag{21}$$

where $\varepsilon(t)$ is the error term, typically assumed to be normally distributed. The main components of the model are:

- **The trend component** $g(t)$. Piecewise logistic and linear growth models are considered.

  The piecewise logistic growth model is

  $$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^T\boldsymbol{\delta})(t - (m + \mathbf{a}(t)^T\boldsymbol{\gamma})))} \tag{22}$$

  $C(t)$ is the expected carrying capacity before saturation of the system at time t, k is the growth rate and m is an offset parameter. Given S changepoints of the trend behavior at times s_j, $\boldsymbol{\delta}$ is a vector of rate adjustments whose components δ_j are the change in rate at time s_j with $j = 1, \ldots, S$. $\mathbf{a}(t)$ is vector whose components a_j are 1 if $t \geq s_j$ or 0 otherwise. Thus, the rate at time t is $k + \mathbf{a}(t)^T\boldsymbol{\delta}$. After adjusting the rate k, the offset parameter m is also adjusted to connect the endpoints of the segments and the correct adjustment at each changepoint j is also computed as γ_j:

  $$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<j}\delta_l}{k + \sum_{l\leq j}\delta_l}\right) \tag{23}$$

  However, the simpler piecewise linear trend model can also be useful:

  $$g(t) = (k + \mathbf{a}(t)^T\boldsymbol{\delta})t + (m + \mathbf{a}(t)^T\boldsymbol{\gamma}) \tag{24}$$

  where the components of $\boldsymbol{\gamma}$ are $\gamma_j = -s_j\delta_j$.

An important feature is the automatic changepoint selection: the model can automatically infer the points s_j where it is more convenient to adapt the trend model from a set of candidates.

- **The seasonality component** $s(t)$. An important feature of this model is the possibility of including multi-period seasonality. In previous models, we assumed a fixed number of periods per seasonal cycle during each training round. Now, we can simultaneously consider weekly and monthly cycles, which may be relevant given the complexity of market dynamics. For any regular period P expected in the time series, arbitrary smooth seasonal effects are approximated via truncated Fourier series with N terms:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \tag{25}$$

The coefficients can be arranged in $\boldsymbol{\beta} = [a_1, b_1, \ldots, a_N, b_N]^T$. Some empirically tested commonly acceptable choices for N could be $N = 3$ for weekly seasonality and $N = 5$ for monthly seasonality. We fit seasonality by building a matrix of seasonality vectors $X(t)$ with the Fourier terms for each t. An example for the monthly component assuming $N = 5$ Fourier terms and a month of 21 trading days is

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{21}\right), \ldots, \sin\left(\frac{2\pi(5)t}{21}\right) \right] \tag{26}$$

Then, the corresponding seasonal component is

$$s(t) = X(t)\boldsymbol{\beta} \tag{27}$$

where $\boldsymbol{\beta} \sim N(0, \sigma^2)$ is a smoothing prior for seasonality.

- **The holidays component** $h(t)$. Here, the term *holidays* has to be understood in a broader sense. It refers to special independent events occurring at particular times that alter the expected dynamics of the time series. The term comes from the fact that the Prophet framework is often applied to business time series affected by holiday periods, but the holiday effects component can be used to model predictable shocks that do not follow a periodic pattern.

  Let D_i be the set of past and future dates for each holiday $i = 1, \ldots, L$. At each time t, we build a matrix of ones and zeros via indicator functions such that $\mathbf{1}(t \in D_i)$ takes the value 1 if time t is during holiday i and 0 otherwise. We also assign each holiday a parameter κ_i which is the corresponding change in the forecast. Then, a matrix of regressors is

$$Z(t) = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)] \tag{28}$$

and then we obtain

$$h(t) = Z(t)\boldsymbol{\kappa} \tag{29}$$

for a smoothing prior for holidays $\boldsymbol{\kappa} \sim N(0, \nu^2)$.

6.2 Implementation

We fit the models using the `Prophet` implementation in the `prophet` library ([29]). As with the other types of models, we design a rolling forecasts algorithm that will allow to retrain the model day after day throughout the year 2022. In this way, we end up with forecasts for the 251 trading days of 2022 through 251 models trained with the most recent data available before each day. Details on the specific implementation of the algorithm can be found in the code in the appendix.

An interesting aspect of this model is that we do not have to choose the specific length of the seasonal cycles because we can add several seasonalities simultaneously. In this case, in order to maintain consistency throughout the work, weekly and monthly seasonality have been specified. Their associated Fourier orders have been set to 3 and 5, respectively. There are some other interesting hyperparameters that are worth optimizing and whose combinations will be tested independently throughout 2022. Namely,

- **Training window size:** The number of days prior to the one where we want to make the prediction. As always, we will consider 4 different training window sizes: 42, 63, 126 and 252 trading days.
- **Changepoint prior scale:** It is related to the flexibility of the trend. We will test three values in different orders of magnitude: 0.005, 0.05 and 0.5.
- **Seasonality prior scale:** It is related to the flexibility of the seasonality. We will test three values in different orders of magnitude: 0.1, 1 and 10.
- **Seasonality mode:** Additive or multiplicative.
- **Days forecasted:** We will perform the retraining process for each day, thus forecasting the price of the next day with each model.

This means that we will test $4 \times 3 \times 3 \times 2 \times 1 = 72$ combinations of values. For each one, 251 models are trained throughout 2022. The output is the list of predictions for each day of the year 2022 along with 95% confidence intervals, the model components and the series of residuals for each of the 251 models associated to each of the 72 combinations of hyperparameters.

6.3 Results

For each combination, we have trained sequentially 251 models to make 251 forecasts. To evaluate the performance, we compute the mean squared error and the proportion of correct predictions of rises and falls. The following results have been obtained for each combination after running the rolling forecast algorithm predicting the next day with each model:

Window size	CPS	SPS	Seasonality	RMSE	Correct predictions
42	0.005	0.1	Additive	9.6900	126 (50.4%)
42	0.005	0.1	Multiplicative	9.6904	127 (50.8%)
42	0.005	1	Additive	9.7079	128 (51.2%)
42	0.005	1	Multiplicative	9.7325	126 (50.4%)
42	0.005	10	Additive	9.6977	129 (51.6%)
42	0.005	10	Multiplicative	9.7103	127 (50.8%)
42	0.05	0.1	Additive	6.1559	119 (47.6%)
42	0.05	0.1	Multiplicative	6.1632	120 (48.0%)
42	0.05	1	Additive	6.1755	117 (46.8%)
42	0.05	1	Multiplicative	5.9467	122 (48.8%)
42	0.05	10	Additive	6.1549	122 (48.8%)
42	0.05	10	Multiplicative	6.5191	136 (54.4%)
42	0.5	0.1	Additive	5.1203	132 (52.8%)
42	0.5	0.1	Multiplicative	5.0126	135 (54.0%)
42	0.5	1	Additive	5.1270	133 (53.2%)
42	0.5	1	Multiplicative	7.1819	142 (56.8%)
42	0.5	10	Additive	5.1666	132 (52.8%)
42	0.5	10	Multiplicative	14.4334	140 (56.0%)
63	0.005	0.1	Additive	11.2710	122 (48.8%)
63	0.005	0.1	Multiplicative	11.2955	123 (49.2%)
63	0.005	1	Additive	11.2824	125 (50.0%)
63	0.005	1	Multiplicative	11.3060	123 (49.2%)
63	0.005	10	Additive	11.2850	125 (50.0%)
63	0.005	10	Multiplicative	11.3003	122 (48.8%)
63	0.05	0.1	Additive	6.4714	115 (46.0%)
63	0.05	0.1	Multiplicative	6.4649	115 (46.0%)
63	0.05	1	Additive	6.4707	120 (48.0%)
63	0.05	1	Multiplicative	5.9123	118 (47.2%)
63	0.05	10	Additive	6.4836	120 (48.0%)
63	0.05	10	Multiplicative	4.6730	127 (50.8%)
63	0.5	0.1	Additive	4.7123	110 (44.0%)
63	0.5	0.1	Multiplicative	4.7193	116 (46.4%)
63	0.5	1	Additive	4.7050	111 (44.4%)
63	0.5	1	Multiplicative	4.7044	128 (51.2%)
63	0.5	10	Additive	4.7074	110 (44.0%)
63	0.5	10	Multiplicative	6.0382	118 (47.2%)
126	0.005	0.1	Additive	11.8341	127 (50.8%)
126	0.005	0.1	Multiplicative	11.7989	124 (49.6%)
126	0.005	1	Additive	11.8621	121 (48.4%)
126	0.005	1	Multiplicative	11.7539	119 (47.6%)

126	0.005	10	Additive	11.8262	130 (52.0%)
126	0.005	10	Multiplicative	11.7711	136 (54.4%)
126	0.05	0.1	Additive	8.0096	116 (46.4%)
126	0.05	0.1	Multiplicative	7.8347	122 (48.8%)
126	0.05	1	Additive	8.0196	117 (46.8%)
126	0.05	1	Multiplicative	7.4191	122 (48.8%)
126	0.05	10	Additive	8.0141	116 (46.4%)
126	0.05	10	Multiplicative	7.3613	115 (46.0%)
126	0.5	0.1	Additive	6.8747	117 (46.8%)
126	0.5	0.1	Multiplicative	6.8830	117 (46.8%)
126	0.5	1	Additive	6.8773	117 (46.8%)
126	0.5	1	Multiplicative	7.0094	122 (48.8%)
126	0.5	10	Additive	6.8769	118 (47.2%)
126	0.5	10	Multiplicative	7.0998	119 (47.6%)
252	0.005	0.1	Additive	13.6750	126 (50.4%)
252	0.005	0.1	Multiplicative	13.6660	129 (51.6%)
252	0.005	1	Additive	13.7007	121 (48.4%)
252	0.005	1	Multiplicative	13.7200	121 (48.4%)
252	0.005	10	Additive	13.6898	125 (50.0%)
252	0.005	10	Multiplicative	13.7002	123 (49.2%)
252	0.05	0.1	Additive	10.7599	126 (50.4%)
252	0.05	0.1	Multiplicative	10.7748	126 (50.4%)
252	0.05	1	Additive	10.8131	131 (52.4%)
252	0.05	1	Multiplicative	10.3349	125 (50.0%)
252	0.05	10	Additive	10.7779	118 (47.2%)
252	0.05	10	Multiplicative	10.1943	129 (51.6%)
252	0.5	0.1	Additive	9.8683	128 (51.2%)
252	0.5	0.1	Multiplicative	9.8700	132 (52.8%)
252	0.5	1	Additive	9.8694	129 (51.6%)
252	0.5	1	Multiplicative	9.7861	130 (52.0%)
252	0.5	10	Additive	9.8667	127 (50.8%)
252	0.5	10	Multiplicative	9.7651	127 (50.8%)

Table 3: RMSE and number of correct predictions of rises and falls throughout the year 2022 for each combination of values.

In terms of the root mean squared error, there are major differences across the different time window sizes. In general, it is suggested to use shorter time windows to favor the adaptability of the model when retraining it day after day. By taking larger training windows, the deviation of the forecasts skyrockets as the model is more sensitive to old information and does not adapt as quickly to new prices that are introduced every day. The other variable with the greatest impact is the changepoint prior scale since it is clear that higher values reduce the error. This

is to be expected since this variable is related to the flexibility of the trend. In our case, the trend can change very quickly over the course of days, so it is necessary to favor the flexibility of the model in this aspect.

As a general observation, the Prophet model might not be particularly suited to this problem, although it was worth giving it a try because of its fast training and the automaticity of the process of obtaining predictions. Unfortunately, it seems that the predictions are sometimes a bit extreme, causing the error to be substantially high, at least compared to other types of models. The number of correct predictions does not seem to be particularly good in most cases either. However, based on the above comments, there do seem to be some specific combinations that stand out positively from the others. If we consider a training window of length 42 days and a changepoint prior scale of 0.5, the error is acceptable in a few of the combinations and precisely the number of correct predictions of rises and falls is substantially better than for most of the other hyperparameter combinations. In particular, consider the combination of 42 training days prior to each prediction, changepoint prior scale 0.5, seasonality prior scale 0.1 and multiplicative seasonality mode. Compared to the baseline where we assume that the best prediction for the next day is the price of the previous day (RMSE = 3.4147) and rises and falls are correctly predicted 50% of the time, we have got an RMSE of 5.0126 and an accuracy of 54.0%. Given that most results in the previous table do not seem to be related to very significant models, this combination seems relatively acceptable.

Specifically, the following forecasts have been obtained for this combination and we compare them with the actual prices.

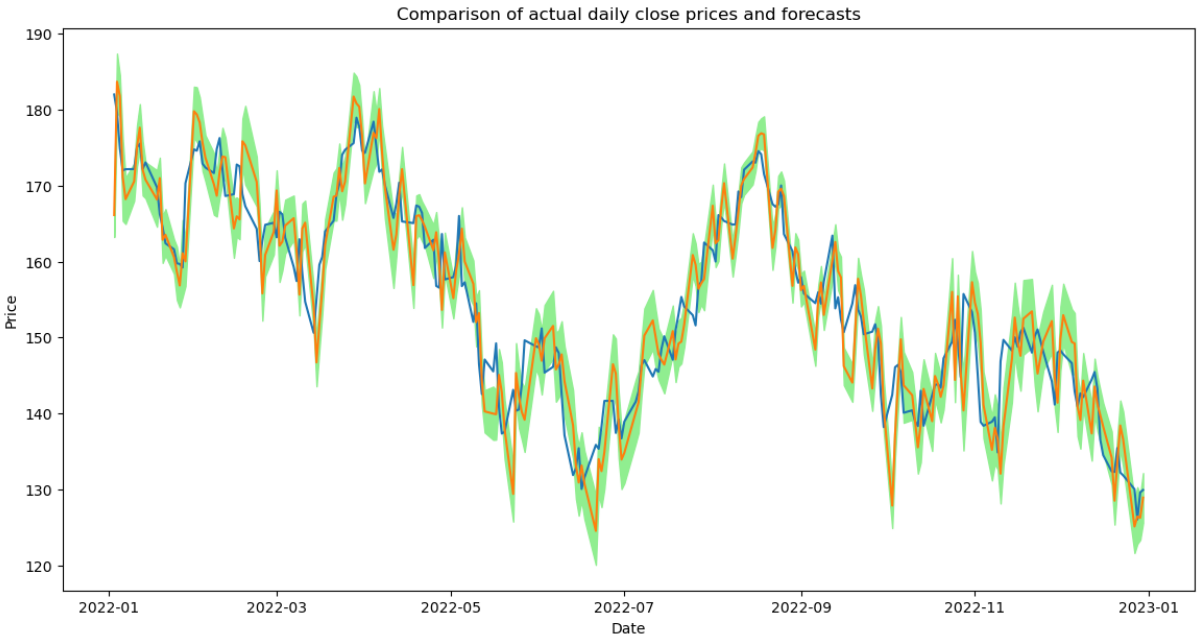

Figure 11: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the Prophet method training with the 42 previous days before each date and using a changepoint prior scale of 0.5, a seasonality prior scale of 0.1 and multiplicative seasonality mode. In green, 95% confidence intervals.

As a comparison with respect to the effect of choosing larger training windows (i.e. training

with older data everyday), see the result that would be obtained with the same configuration, but selecting the 252-day window.

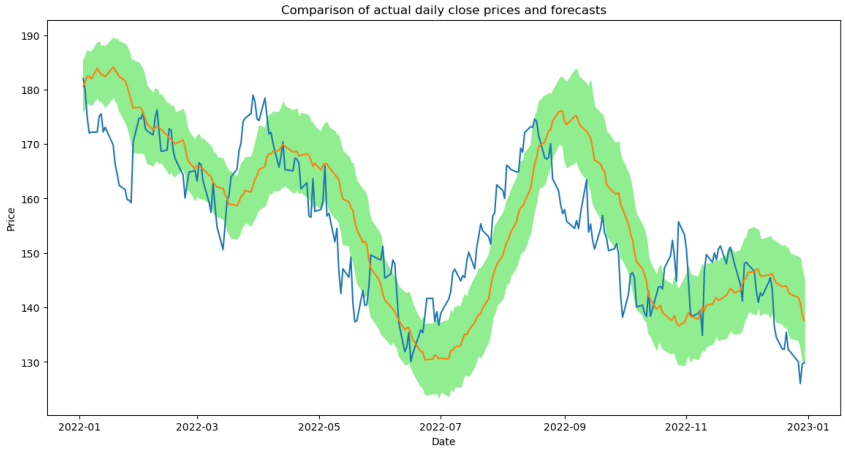

Figure 12: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the Prophet method training with the 252 previous days before each date and using a changepoint prior scale of 0.5, a seasonality prior scale of 0.1 and multiplicative seasonality mode. In green, 95% confidence intervals.

As a comparison with respect to the effect of choosing a smaller changepoint prior scale (i.e. less flexibility of the trend), see the result that would be obtained with the same configuration, but selecting a changepoint prior scale of 0.005.

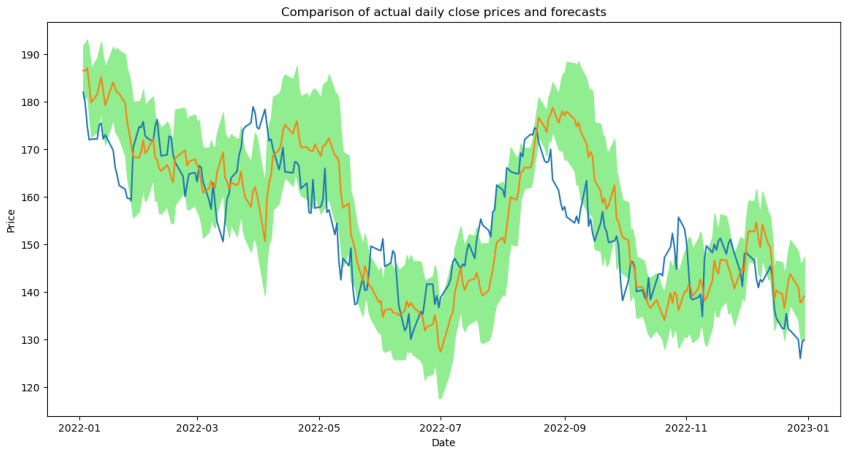

Figure 13: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the Prophet method training with the 42 previous days before each date and using a changepoint prior scale of 0.005, a seasonality prior scale of 0.1 and multiplicative seasonality mode. In green, 95% confidence intervals.

In the most extreme case, with a training window size of 252 days and changepoint prior scale of 0.005, observe the forecasts.

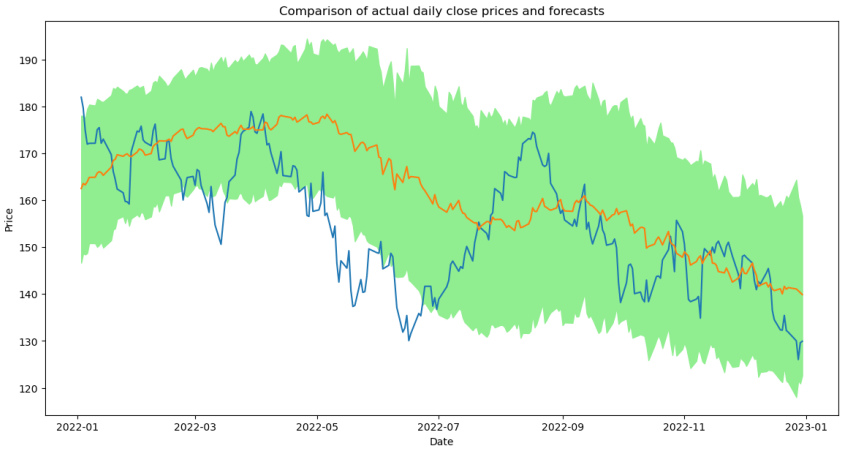

Figure 14: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the Prophet method training with the 252 previous days before each date and using a changepoint prior scale of 0.005, a seasonality prior scale of 0.1 and multiplicative seasonality mode. In green, 95% confidence intervals.

In general, Prophet is a predictive method that works best with series with a strong seasonal component. It is also a model that is resistant to outliers and, as we train with more data, it performs longer-term smoothing without incorporating so actively the rises and falls of the market on a day-to-day basis. However, some selections of hyperparameters may allow to exploit the capabilities of the model in handling trend shifts.

7 ARIMA models

ARIMA stands for *AutoRegressive Integrated Moving Average*, an acronym where we distinguish three parts: AR, I, MA. We will examine their contributions step by step.

7.1 Theory

Consider a univariate time series $\{y_t\}$. In a standard linear regression problem, the target variable is modeled as a linear combination of predictors. The models of the ARIMA family are based on linear functions of several past observations and random errors.

- The idea of an AutoRegressive (AR) model is to describe the present value of the time series y_t as a linear combination of its past values, along with a white noise term and possibly an independent term. In general, an autoregressive model that depends on p lagged values of y_t is called AR(p) and it can be expressed as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t \tag{30}$$

  where c is an intercept term, $\phi_1, \ldots, \phi_p$ are the p parameters associated to the p lagged values of y_t and ε_t is a white noise series with mean zero and variance σ^2.

  As an example, the simple AR(1) model with intercept is given by the equation $y_t = c + \phi_1 y_{t-1} + \varepsilon_t$. If $\phi_1 = 0$, this is just a white noise process. If $\phi_1 = 1$ and $c = 0$, this is called a *random walk*. If $\phi_1 = 1$ and $c \neq 0$, this is a *random walk with drift*.

  Consider the backward shift operator B such that $By_t = y_{t-1}$. Then, the AR(p) model can be written as:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p) y_t = c + \varepsilon_t \tag{31}$$

  If any of the roots of the characteristic equation

$$1 - \phi_1 x - \phi_2 x^2 - \ldots - \phi_p x^p = 0 \tag{32}$$

  has a modulus smaller than 1, then the process has a unit root and it is non-stationary. In general, we restrict autoregressive models to stationary data. For example, a necessary condition for an AR(1) process to be stationary is that $|\phi_1| < 1$.

  For a pure AR(p) process, the partial autocorrelation function cuts off at lag p (in a real estimation of an AR(p) model, it would not vanish completely, but the PACF would show the first not significant value at lag p, which is a model selection criterion).

- The idea of a Moving Average (MA) model is to describe the present value of the time series y_t as a weighted moving average of past forecast errors and possibly an independent term. In general, a moving average model that depends on q lagged values of ε_t is called MA(q) and it can be expressed as:

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q} \tag{33}$$

where c is an intercept term and $\theta_1, \ldots, \theta_q$ are the q parameters associated to the q lagged values of ε_t, which is a white noise series with mean zero and variance σ^2.

Pure AR models are always invertible. However, under certain conditions of invertibility, we can express MA(q) processes as AR(∞) processes. The simple MA(1) model without intercept is given by the equation $y_t = \varepsilon_t + \theta_1\varepsilon_{t-1}$. This process is invertible when $|\theta_1| < 1$.

Consider the backward shift operator B such that $By_t = y_{t-1}$. Then, the MA(q) model can be written as:

$$y_t = c + (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q)\varepsilon_t \tag{34}$$

A finite MA model is always stationary.

For a pure MA(q) process, the autocorrelation function cuts off at lag q (in a real estimation of an MA(q) model, it would not vanish completely, but the ACF would show the first not significant value at lag q, which is a model selection criterion).

An ARMA model is a combination of some of the previous ideas to obtain a parsimonious model for a stationary time series. Combining the equations for AR(p) and MA(q) models, an ARMA(p, q) model is given by the equation

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q} \tag{35}$$

In polynomial form:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)y_t = c + (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q)\varepsilon_t \tag{36}$$

A general ARMA(p, q) requires the estimation of $p + q + 2$ parameters (p parameters for the autoregressive lags, q parameters for the moving average lags, 1 parameter for the variance of ε_t and 1 intercept c) or just $p + q + 1$ if the intercept is not considered.
Obviously, ARMA($p, 0$) is equivalent to AR(p) and ARMA($0, q$) is equivalent to MA(q).
In contrast to the AR and MA models, the ACF and PACF are not informative enough for an accurate selection of p and q simultaneously (although they might give an approximation) and we resort to the selection of a model with minimum AIC.
The ARMA models seen so far are designed for the modeling of stationary time series, without unit roots. However, it is clear that a large number of the time series of interest will have a trend component or will exhibit some type of non-stationary behavior over time. An ARIMA model is capable of dealing with non-stationary time series through the consideration of as many differences as necessary to make the series stationary. Considering ∇^d, the lag-1 difference operator of order d which applies d differences and that can also be written as $(1 - B)^d$, an ARIMA(p, d, q) model is

$$(1 - B)^d y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q} \tag{37}$$

The ARIMA family of models easily allows the inclusion of exogenous variables as extra linear terms. Consider r covariate series $x_{1,t}, \ldots, x_{r,t}$. An ARIMAX(p, d, q) is of the form:

$$\begin{aligned}(1-B)^d y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} \\ + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \\ + \beta_1 x_{1,t} + \ldots + \beta_r x_{r,t}\end{aligned} \quad (38)$$

However, it is possible for a series to be non-stationary, not because of the presence of a trend component, but because of a seasonality component. This family of models is prepared to deal with the effects of seasonality, knowing that this component can be eliminated through seasonal differences. A non-seasonal ARIMA model requires the specification of the number of autoregressive lags p, the number of ordinary differences of the series d and the number of moving average lags q. In order to take into account the seasonal component, we will now consider a number of seasonal autoregressive lags P, a number of seasonal differences D, a number of seasonal moving average lags Q and the number of periods per seasonal cycle m. A SARIMAX$(p, d, q)(P, D, Q)_m$ model in polynomial form is given by:

$$\begin{aligned}(1-\phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(1-\psi_1 B^m - \psi_2 B^{2m} - \ldots - \psi_P B^{Pm})(1-B)^d (1-B^m)^D y_t = \\ = c + (1+\theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q)(1+\zeta_1 B^m + \zeta_2 B^{2m} + \ldots + \zeta_Q B^{Qm})\varepsilon_t + \beta_1 x_{1,t} + \ldots + \beta_r x_{r,t}\end{aligned} \quad (39)$$

7.2 Implementation

The calculation of the parameters of the SARIMA models has been carried out using the `auto_arima` implementation in the `pmdarima` package ([30]). Following the procedures described in the methodology section, we will implement an algorithm to obtain rolling forecasts retraining day after day over the course of the year 2022. Details on the specific implementation of the algorithm can be found in the code in the appendix.

Given the good routines already implemented for selecting the order of the models by minimizing the AIC, the algorithm will find the best choices. However, the orders of the differences have been manually specified since it has been found that one difference ($d = 1$) and one seasonal difference ($D = 1$) are more than enough to turn the series into a stationary one (it can be checked with appropriate tests such as the ADF test). It has been deemed convenient to tune the following variables:

- **Training window size:** The number of days (prior to the day we want to forecast) that we use to train the model. As in all models considered in this document, we will test training windows of size 42 days, 63 days, 126 days and 252 days to see how old the training data should be at most.
- **Seasonal period:** Because the seasonal component is difficult to identify, it is not very clear how many periods we should consider in a seasonal cycle. In a time series of daily prices, it is possible that there are behaviors with weekly (5 days) or monthly (21 days)

periodicity. Therefore, this is another hyperparameter that should be tuned. However, the greater the number of periods per cycle, the more training time is required.

- **Days forecasted:** We will focus on making predictions for the next day after the training data.

This implies that we have to test $4 \times 2 \times 1 = 8$ combinations of values. This is done for each round of 251 models that are trained successively throughout 2022. For each of the 8 combination of values, the output is the list of predictions for each day of the year 2022 along with 95% confidence intervals, the series of residuals for each of the 251 models, the list of orders and seasonal orders of the fitted SARIMA models each day and the corresponding values of the fitted parameters.

7.3 Results

The procedure will initially consist of finding the best hyperparameter configuration without exogenous variables by carrying out the study described in the previous subsection.
For each combination of training window size and length of each seasonal cycle, we evaluate the RMSE and the number of correct predictions of rises and falls in the closing prices. Below, we see the results.

Time window size	Seasonal periods	RMSE	Correct predictions of rises/falls
42	5	4.0635	124 (49.6%)
42	21	5.0548	132 (52.8%)
63	5	3.8536	133 (53.2%)
63	21	4.4787	140 (56.0%)
126	5	3.8369	125 (50.0%)
126	21	3.9585	142 (56.8%)
252	5	4.0463	122 (48.8%)
252	21	3.6956	138 (55.2%)

Table 4: RMSE and number of correct predictions of rises and falls throughout the year 2022 for each combination of values.

Interesting patterns seems to appear: the RMSE tends to get a bit better when we choose larger training windows and seems to be worse for the combinations with 21 seasonal periods with the exception of the last time window size. On the other hand, looking at the number of correct predictions of rises and falls, we notice that the numbers are better when we choose 21 seasonal periods instead of 5. Trying to maintain an adequate balance between both metrics, we could say that the combination with training window size 126 and 21 seasonal periods is pretty acceptable among the combinations that have been analyzed (best number of correct predictions and moderately low RMSE). We pick this configuration for the rest of this section.

Compared to the baseline (assume that today's closing price is the same as yesterday's and the rises and falls are correctly predicted 50% of the time), the SARIMA models trained with the 126 previous days before each day and seasonal cycles of 21 days produce an RMSE of 3.9585,

whereas the baseline stands at 3.4147. Fortunately, we will soon be able to correct this deviation in the RMSE by adding covariates. As for the proportion of correct predictions of rises and falls, the 56.8% accuracy can be deemed acceptable for the standards of the problem.

Of course, each model has its own set of parameters. In the case of the SARIMA$(p, d, q)(P, D, Q)_m$ models, this depends on the orders and seasonal orders that indicate the number of lags to be considered. In this case, we fixed $d = 1$ and $D = 1$, the seasonal order m is varied and p, q, P and Q are found by minimizing the AIC. Judging by the orders of the optimal models (with respect to the AIC), not excessively complex but definitely not trivial models have been obtained. Among the 251 models fitted for all days:

- For p:
  - 206 models had $p = 0$.
  - 41 models had $p = 1$. The corresponding value of the parameter oscillates between -0.8524 and 0.2359 depending on the model associated with each day.
  - 3 models had $p = 2$. The first coefficient oscillates between 0.6508 and 0.6621 and the second coefficient oscillates between -0.9722 and -0.9650 depending on the model associated with each day.
  - 1 model had $p = 3$. The first coefficient is 0.1790, the second coefficient is -0.1303 and the third coefficient is -0.7438.
- For q:
  - 172 models had $q = 0$.
  - 75 models had $q = 1$. The corresponding value of the parameter oscillates between 0.1129 and 0.9658 depending on the model associated with each day.
  - 3 models had $q = 2$. The first coefficient oscillates between -0.5670 and -0.5650 and the second coefficient oscillates between 0.8550 and 0.8702 depending on the model associated with each day.
  - 1 model had $q = 3$. The first coefficient is -0.1997, the second coefficient is -0.0805 and the third coefficient is 0.9043.
- 13 models incorporated an intercept that oscillates between -0.3841 and 0.7059 depending on the model associated with each day.
- The variance of the residuals (σ^2) oscillates between 5.6788 and 15.2989 depending on the model associated with each day.
- For P:
  - 94 models had $P = 0$.
  - 17 models had $P = 1$. The corresponding value of the parameter oscillates between -0.2776 and -0.0305 depending on the model associated with each day.

- 140 models had $P = 2$. The first coefficient oscillates between -0.8603 and -0.0867 and the second coefficient oscillates between -0.5607 and -0.0473 depending on the model associated with each day.

- For Q:
  - 110 models had $Q = 0$.
  - 119 models had $Q = 1$. The corresponding value of the parameter oscillates between -0.8069 and -0.4058 depending on the model associated with each day.
  - 22 models had $Q = 2$. The first coefficient oscillates between -0.9505 and -0.7842 and the second coefficient oscillates between 0.0411 and 0.2580 depending on the model associated with each day.

The combinations of orders appear in clusters, indicating that the fit of the models is not purely random, and in areas of similar data, similar models are fitted, as expected. In many cases, the order of the ARMA part is trivial since the model is not able to capture any valuable relation with previous lags. In others, it is considered relevant to include some low order lag in the AR or MA part. In the case of the parameters of the seasonal part, a higher number of iterations have been detected in which models with at least one lag in the seasonal autoregressive part or in the seasonal moving average part are fitted.

We plot the forecasts and their confidence intervals along with the actual values of the prices for comparison.

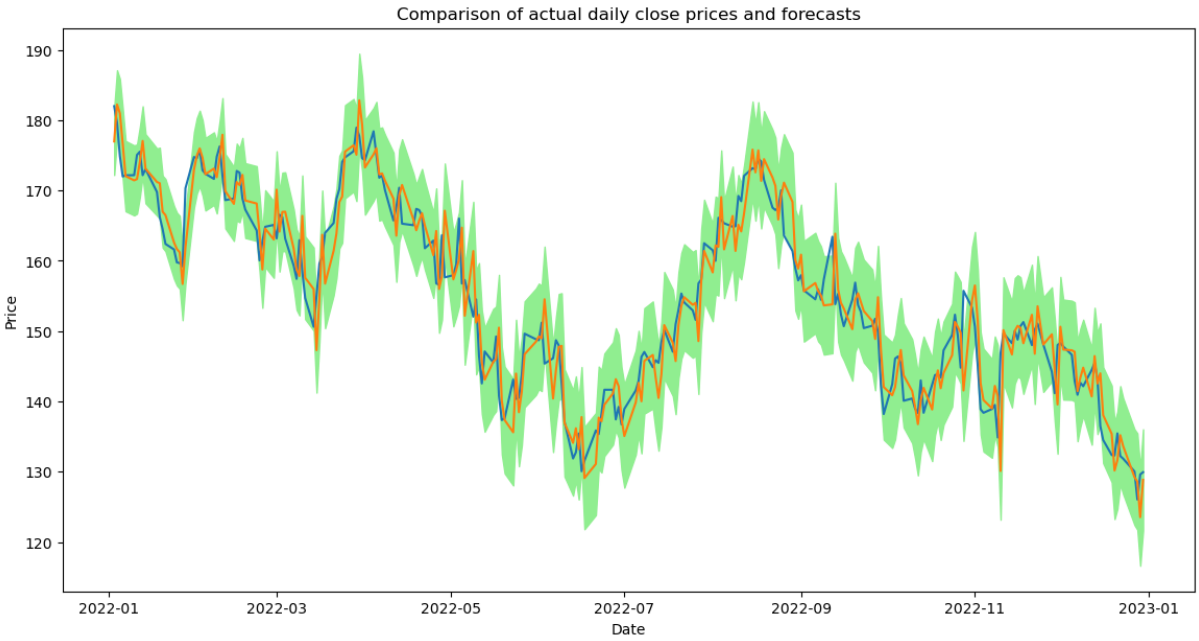

Figure 15: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the SARIMA method training with the 126 previous days before each date and assuming seasonal cycles of period 21. In green, 95% confidence intervals.

As a reminder, this is the result of 251 models sequentially trained to make a prediction for the next day, which serves as a realistic simulation of the predictions that a trader could obtain for

each day of a year. The forecasted prices are not too far away from the actual prices and the rolling forecast algorithm provides predictions that adapt well over time when retrained on a day-to-day basis.

We also consider the interesting approach of a finer definition of the price movements. A price is said to have experienced a lateralization if its percentile variation (percentile return) with respect to the previous day is between -0.5% and 0.5%. These cases will correspond to class 0. If there is a fall that implies a return lower than -0.5%, we will consider that these days belong to class 1. In the case of rises involving returns greater than 0.5%, this is class 2. From the above predictions, the construction of these classes is straightforward. We also construct them for the time series of actual prices. The result is summarized in the following confusion matrix.

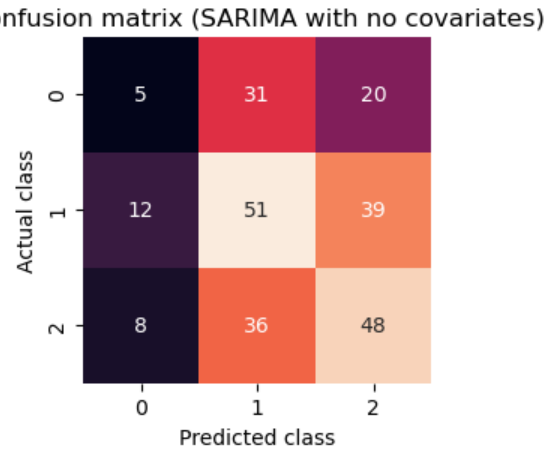

Figure 16: Confusion matrix of the SARIMA model with no covariates comparing the classes of the actual daily closing prices and the classes of the forecasts. Class 0: Lateralizations (returns between -0.5% and 0.5%). Class 1: Returns lower than -0.5%. Class 2: Returns greater than 0.5%.

Looking at the diagonal, we see that the lateralizations (class 0) are not really captured. On the other hand, classes 1 and 2 are a bit better classified, although the limitations of the model must be acknowledged, since the base accuracy shown in the table is not miraculous.

Can we do any better? SARIMA models include the possibility of adding exogenous variables to the model in a very natural way. Using the combination of parameters that we have selected for the previous analysis as the best, we are going to refit the models, but this time adding the covariates to see if it is possible to improve the RMSE and the number of correct predictions of rises and falls. In particular, these covariates are the opening price of the AAPL stock, the opening price of the SPY stock and the volume of AAPL shares traded each day. This implies the addition of a new parameter in the model for each of the exogenous variables. On many days, we will find that the value of these parameters is no more significant than if they were null (this is checked with the t-test that compares the value of the parameter with its standard error), but when they are significant, they can be very valuable regressors.
In this case, the new RMSE resulting from repeating with covariates the procedure that was

carried out without covariates to discover an optimal parameter configuration has been found to be 3.4139. This error is much better than in the case without covariates and even beats the defined baseline. On the other hand, the number of correct predictions rises and falls has now become 148 (59.2%), which is also a nice improvement over the models without covariates.

The new forecasts are compared to the actual prices in the following plot:

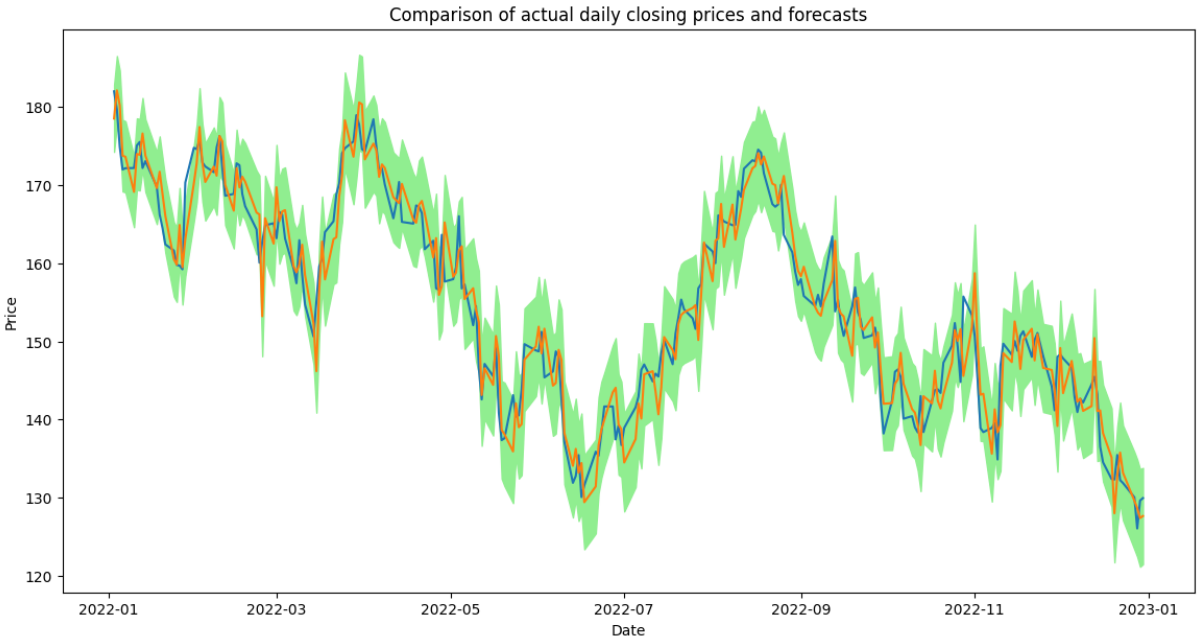

Figure 17: In blue, actual daily closing prices for Apple in 2022. In orange, rolling forecasts from the SARIMA method with covariates training with the 126 previous days before each date and assuming seasonal cycles of period 21. In green, 95% confidence intervals.

Preparing again a three-class classification problem (class 0: lateralization, class 1: substantial fall, class 2: substantial rise), we find the following confusion matrix:

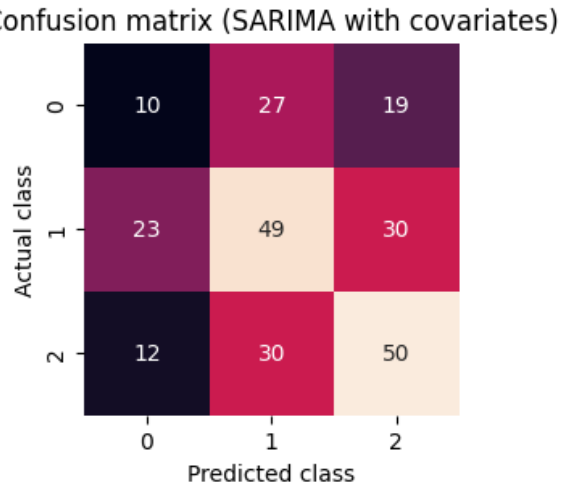

Figure 18: Confusion matrix of the SARIMA model with covariates comparing the classes of the actual daily closing prices and the classes of the forecasts. Class 0: Lateralizations (returns between -0.5% and 0.5%). Class 1: Returns lower than -0.5%. Class 2: Returns greater than 0.5%.

Compared to the previous confusion matrix, we now observe a significant shift toward the more frequent prediction of lateralizations. This is obviously good for the prediction of class 0. However, we could also say that the prediction of class 1 and class 2 have got a bit better. Even though their corresponding diagonal elements remain at a quite similar value, class 2 and class 1 respectively were predicted significantly fewer times in favor of the more conservative prediction that represents class 0, thus reducing the severity of the error.

Recall that the value of the covariates must be known in the day where we will make a prediction before making such forecast. In the code shown in the appendix, these last results are obtained with a function that has been named `rolling_sarima_covariates_1`. This function assumes that the forecasts are made after the market opens, so the actual opening prices of AAPL and SPY are known for the day of the forecast of AAPL closing price. This is useful for intraday trading, but of course, we will also be interested in forecasting well in advance. Specifically, being able to make predictions with covariates from the previous day. This has already been taken into account and therefore a second function `rolling_sarima_covariates_2` is proposed in the appendix. The difference is that the training is carried out with the opening prices as in the first function, but when predicting, it is assumed that a good approximation of the opening price of AAPL and SPY are their corresponding closing prices of the previous day (although they are not exactly the same, since the companies continue their activities even if the stock exchange is closed).

7.4 Natural extensions

Beyond the ARIMA family methods studied, it would be possible to generalize some aspects or to make more complete models that could be useful in some cases. A brief summary is described below.

7.4.1 VARMAX models

The letter V stands for *vector*. In the previous sections, we have focused on the approach of forecasting future values of one time series with the help of additional exogenous variables. Nevertheless, within the formalism of the ARMA models, it is possible to generalize the equations to model the dynamics of several stationary time series simultaneously and to make joint forecasts of several target time series (and if necessary, we could also add extra exogenous variables). This motivates the development of VAR, VMA, VARMA models and the corresponding extensions to account for seasonality and exogenous variables.
A general VARMAX(p, q) is a generalization of the univariate ARMAX models presented before. A VARMAX(p, q) model is defined by the following equation:

$$\begin{aligned} Y_t = C + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \ldots + \Phi_p Y_{t-p} \\ + \mathcal{E}_t + \Theta_1 \mathcal{E}_{t-1} + \Theta_2 \mathcal{E}_{t-2} + \ldots + \Theta_q \mathcal{E}_{t-q} \\ + \mathcal{B} X_t \end{aligned} \quad (40)$$

At each time t, Y_t represents a state as a vector of n target variables, $\mathcal{E}_t$ is a vector of n errors, X_t is a vector of r exogenous variables, C is a vector of n intercepts, each Φ is a matrix of $n \times n$ autoregressive parameters, each Θ is a matrix of $n \times n$ moving average parameters and $\mathcal{B}$ is a matrix of $n \times r$ exogenous variable parameters.
An interesting point of a VARMA model (without exogenous variables) is that it is possible to make simultaneous forecasts for each individual series in the vector without additional information in the time steps that we want to forecast. In contrast, an ARMAX model requires the knowledge of the values of the corresponding exogenous variables in the time steps that we want to forecast before making the forecast (sometimes, we may have that information if the covariates are well-known or it is compulsory to estimate it with the necessary assumptions or with secondary models). Thus, the VARMA and ARMAX formulations are not equivalent.
However, the estimation of general VARMAX models may carry robustness problems and there may be identification issues for many of the parameters. The estimation takes a long time, they may be hard to regularize and they are predisposed to convergence issues. Since our objective is to work with price series of specific companies and complement them with external series, and many parameters even for the most basic specifications were not significant (using the popular implementation in the `statsmodels` library for Python), this line of research has not been continued.

7.4.2 ARMA-GARCH models

GARCH stands for *Generalized AutoRegressive Conditional Heteroscedasticity*. The reason why it is relevant to discuss them here is because of the existence of hybrid ARMA-GARCH models. Unlike ARMA models with which we seek to obtain forecasts of the mean of the time series, a GARCH model is used to forecast the standard deviation (or volatility in economic terms). This is, how much the value of an asset is expected to fluctuate over a given period. Previously, we have been able to observe how the economic series present periods of mild volatility and occasionally periods of very pronounced volatility due to economic instability. This introduces

heteroscedasticity into the data. The standard way to deal with heteroscedasticity prior to using models is to transform the data, typically by taking logarithms. However, this may not cancel the heteroscedasticity completely and we are often interested in modeling changes in variance. Changes in volatility do not always occur over long periods of time, but often come in clusters. The occurrence of these clusters may be heavily subject to specific external events that generate uncertainty, but the past behavior of the series may provide some information for the prediction of highly volatile periods. This analysis is especially relevant in the economic domain where changes in volatility are directly related to changes in economic stability. Investing in times of high volatility involves a number of risks, so it is an important part of price series modeling.
A general GARCH(r, s) model can be described as

$$y_t = \mu_t + \varepsilon_t \tag{41}$$

$$\varepsilon_t = \sigma_t e_t \tag{42}$$

$$\sigma_t = \sqrt{\omega + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_r \varepsilon_{t-r}^2 + \beta_1 \sigma_{t-1}^2 + \ldots + \beta_s \sigma_{t-s}^2} \tag{43}$$

Here, μ_t is the *mean equation* which can be a constant or another autoregressive process and σ_t is the *volatility equation*. e_t is a sequence of IID random variables with mean 0 and variance 1 (typically, normal or Student's t distributions).
Determining the optimal hyperparameters in a GARCH model is somewhat more complicated than in an ARMA model. Nevertheless, the use of Akaike Information Criterion and a careful examination of the significance of the parameter estimates usually results in good models. Another heuristic used by some authors is to look at the values of the partial autocorrelation function of the squared values in the first lags and select the GARCH model hyperparameters according to the number of lags for which such function takes significant values. In any case, low order GARCH models are used in most applications.
An ARMA(p, q)-GARCH(r, s) process consists of the specification of an ARMA model as conditional mean equation, a GARCH model as conditional variance equation and a distribution for the standardized residuals. It is possible to estimate the parameters of the ARMA and GARCH parts of an ARMA-GARCH model simultaneously. However, another common technique that usually works well in practice is to fit both models sequentially. That is, we fit an ARMA model on the original series to remove *the linear part* and then fit the GARCH model on the series of residuals with volatility clusters.
To see how these models work, we have experimented with some implementations using the `arch` library in Python, which is the most popular for GARCH models. Nevertheless, since we are focusing on the modeling of the prices themselves rather than on their volatility, we have concentrated on the optimization of the SARIMA models described above.

8 Conclusion and further work

In this thesis, we have presented a self-contained analysis of relevant theory of time series analysis and its applications to financial data series of the stock market. Some models have been discussed and applied to real data to illustrate the main steps of the methodology for economic time series analysis and predictions in financial markets. We have looked at the peculiarities of each model, the ease or difficulty of optimizing them, their readiness to deal with additional features, and their strengths and weaknesses.

There are a myriad of models that could be used for stock forecasting. In this work, we have analyzed a selection of models with a special relevance in the field of statistical models for time series. The simpler a model is, the easier it is to interpret its parameters, to modify its structure and to fit it, but it also may not have sufficient capacity to provide accurate results. On the contrary, a complex model is possibly better prepared to capture complex patterns and obtain better forecasts, but its manipulation is more sophisticated, it is harder to fully optimize it and its computational cost is more challenging. The approach with classical statistical models is well adapted to the treatment of time series, but we cannot guarantee with absolute confidence the reliability of the results since it is difficult to capture a stable trend and seasonality, on which the quality of the results greatly depends. There are also many modern models that are expected to perform better than classical statistical models, but complex models are not always able to deliver better statistically significant results and they are also less explainable. A key idea is that a simple model may not necessarily perform worse than a complex model under certain circumstances.

Regarding the specific results of this thesis, we have presented both satisfactory and unsatisfactory honest results relative to the standards for this problem. Three models have been analyzed, showing a variety of options for their configuration. Then, for each one, we picked a configuration that seemed fine and displayed the specific results of the rolling forecasts throughout one year. Not all models perform equally well. The Holt-Winters model proposes a relatively simple training process and some results that could be considered remarkable in some sense. The Prophet model is an excellent time series model, but applied to this problem it has not seemed to provide very significant results. The SARIMA model is a classic of time series modeling and offers some interesting features. Even more interesting is the possibility of adding exogenous variables, which has been explored and it has been possible to improve the original results. The addition of covariates has been very fruitful in the case of the SARIMA model, which opens the door to the study of many other possible social and economic variables that influence market dynamics.

As the British statistician George E. P. Box said, *all models are wrong, but some are useful.* Are these models actually useful? We could defend both points of view. Answering the question in the negative, the community discusses some theories such as the random walk hypothesis stating that stock prices evolve unpredictably or the efficient market hypothesis stating that stock prices reflect all available information and it is impossible to outperform the overall market with stock picking or market timing, except in very specific and extraordinary cases. Answering the question in the positive, some significant results are reported in the literature and interest in technical

analysis of the market has increased in recent years. There is a greater interest in the use of statistical techniques and more quantitative analysts are being hired to build predictive models and analyze more return predictors.

The inherently erratic behavior of financial market time series makes it difficult to develop stable models with high predictive power. Market dynamics are constantly changing due to numerous factors and we have to restrict ourselves to using data that is not too far away from the dates we want to forecast. For time series that do not have a completely stable and persistent trend or seasonality, it is difficult to guarantee the validity of the forecasts. It is also hard to identify those economic or social characteristics that give rise to certain stock market behaviors and how to preprocess them adequately to put them in the right scale and time granularity. Although some patterns are known, this is still an active area of research. In general, it is unrealistic to think of the possibility of obtaining very general methods for any time and stock with low prediction errors and accurate forecast rates of rises and falls of more than 60%. There are studies with very powerful models that include a wide range of information across dozens of variables (for example, see [7]), yet it is still unfeasible to develop a general framework to fully understand and capture market dynamics in real time. An important part of mathematical modeling is the understanding of its limits and the limitations of the models are noticeable. This work is simply a demonstration of the results (both those that are significant and those that are not) that can be obtained by following a well-defined methodology and observing the particularities of each model. In any case, the main contribution of this thesis is not the specific results, but the methodology and the definition of a workflow idea that makes sense. Any modification that is deemed convenient (such as, for example, considering a smaller test set), is a matter of indicating it in the code.

In any case, although market forecasting is a challenge for which no definitive solution can be given, it is an interesting mathematical problem to tackle and to see to what extent mathematical modeling can provide information of interest. Indeed, the modeling of financial markets is a very complex subject with a considerable number of challenges. As has been noted during the completion of this project, a relevant challenge in computational finance is the computation time. There is a high computation cost involved in the process of building models training with sufficiently diverse parameter grids and validating them over sufficiently large time horizons. There is always room for improvement in any model by further altering parameters or exploring alternative formulations, but we have attempted to do a sufficiently thorough and honest analysis of the idiosyncrasies of each model. In this project, we have tried to forecast daily series because it is one of the problems with the most practical utility since an investor is usually interested in predicting the state of the market at the end of the day. However, it is also a more complicated problem to solve, since it is difficult to create models that capture the most relevant information on a daily basis without noise. Another interesting problem would be to aggregate the price data to work with weekly or monthly price series, although this significantly decreases the amount of recent data to work with and reduces the number of possible predictions.

It is difficult to assess the significance of certain results and whether the methods presented would generalize well to any time since past performance is not a guarantee of results for all times. Also, some forecasting methods tend to be more conservative in prediction by offering forecasts close to the previous day's value when in doubt. Therefore, on average, it gives the feeling that

their results are better than in models that offer more extreme predictions in the sense that the correct predictions are more accurate and the failures are more misleading. However, this does not mean that a specific model leads to better results than others in every situation. There is not a set of models or a systematic combination of hyperparameters that works well for all stocks. Some methods may perform well for some periods, but not for others, and it can be impossible to determine when to use one model or another for actual forecasts without actual known test data.

It should be noted that accurately predicting specific financial market daily quantities and their rises and falls remains an open problem. To solve this problem, much research has been carried out in recent years and the usefulness of the whole array of predictive methods offered by modern statistics has been tested. Unfortunately, advances in this area of finance are not always readily accessible. Remarkably outstanding and profitable methods developed by private institutions are commonly not freely published in scientific journals. In addition, privileged knowledge of some social and economic variables by certain individuals may give them a crucial comparative advantage, especially with certain variables which are difficult to predict, are hardly accessible for the general public and may significantly help in predicting final stock prices. Still, some specific patterns of market dynamics might be inherently unpredictable for everyone, even with the inclusion of a large number of seemingly relevant covariates. In any case, the development of new methods and the influence of new predictors continues to be an active area of research with much room for future work. Numerous articles with state-of-the-art models continue to be published using the most complete information available and increasingly sophisticated methods of data analysis.

In reality, economic analysis has two parts: fundamental analysis and technical analysis. This project has focused on the technical analysis of statistical models, but the fundamental analysis performed by economists and other experts in the field should not be overlooked in order to truly assess the state of the market. It is a field that requires extensive domain knowledge. The right mathematical tools and good training on the subject help make better decisions, but one always has to proceed with care and knowledge about fundamental analysis and market risks. From modern portfolio theory, there are a number technical investment indicators that should be taken into account when deciding to buy and sell stocks to evaluate the potential risks and returns simultaneously. In addition, bearing in mind that markets are always exposed to unexpected catastrophic events, under no circumstances does this thesis encourage the reader to engage in specific transactions with specific models. There is no definitive infallible method for investing in the stock market without risk. However, a careful mathematical analysis can lead to profitable results such as in the case of many companies that are specifically dedicated to the development of methods for stock market predictions. While it may not generally possible to obtain price rise and fall accuracy rates significantly higher than 50%, the underlying rationale is to develop models that, in the long run, produce profits through models with forecasts of prices sufficiently close to actual prices and rise and fall accuracy rates slightly higher than 50%, thus overcoming potential losses.

A complete model should be powerful enough and should take into account as much relevant information as possible. Therefore, possible improvements to the statistical techniques presented here include experimenting with more types of models and predictors.

Regarding models, one of the most talked about family of models in recent times are the Transformers. There are numerous successful cases of this type of model in the Natural Processing Language (NLP) area. However, its distinctive self-attention mechanism can also be used for time series forecasting. Some modern implementations like the Informer [25] or the Autoformer [24] could be good candidates for developing even more powerful predictive methods.
Using historical prices to detect patterns and trends is a good way to try to the predict the future dynamics of the prices. However, numerous unforeseen circumstances may arise and we will always need further market analysis before making any investment. To give a specific example of a way of integrating more external information, NLP can help to obtain information from financial news. Sentiment analysis of financial news could give an idea of whether the value of a company will increase or decrease based on real-time information about the actions, successes, failures, future expectations and hype around the company. Therefore, some news-based variables can potentially be relevant since they are expected to have a close relation to the dynamics of the prices (rises and falls, as well as their magnitude and the strength of volatility). How to accurately design such a metric and the choice of accurate financial information is another problem that is being actively researched. Some recent attempts to include financial news in stock forecasting with LSTM-based networks are [13] and [23].
Another issue that is often discussed is the problem of low frequency covariates. When trying to forecast daily prices (or the problem would be even more serious when trying to predict prices with higher frequency), we may want to add information for which we only have one data point every week, month or quarter. This is very common with some macroeconomic variables that we may want to add to the models. However, this can be problematic as many models require all variables to be sampled at the same frequency. A solution that could be worth analyzing in the future is the application of temporal disaggregation methods (see [18]). These are methods to turn low frequency time series into higher frequency series, where the sum, the average, the first or the last value of the resulting high frequency series is consistent with the low frequency series. In addition, other higher frequency economic indicators can be considered so that the disaggregation incorporates information of more value than just a regular interpolation. Another model specially designed to deal with series in different frequencies is the MIxed DAta Sampling (MIDAS) regression. It was originally introduced as a method to make forecasts in low frequency time series with the help of high frequency covariates. However, a Reverse MIDAS model was later introduced in [3] to deal with the opposite problem: forecasting a high frequency time series with the help of low frequency predictors. Other examples of MIDAS-based models with indicators sampled at a lower frequency than the main series include multi-indicator multi-output methods like in [16].
From a more personal point of view, this thesis has been a very valuable opportunity to learn a lot about the interesting world of stock markets, explore a series of statistical methods and improve my programming skills in Python. The help and experience from the StockFink staff have also been very helpful in understanding which details were most interesting to explore and analyze. Since the applications of statistical models are becoming more and more notable, the knowledge acquired in this work will undoubtedly be very useful in the future.

References

[1] Asokan, M. (2022). *A study of forecasts in Financial Time Series using Machine Learning methods: Traditional vs Machine learning approach*. Master's thesis, Linköping University.

[2] Brockwell, P. J., & Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. 3rd edition. Springer.

[3] Foroni, C., Guérin, P., & Marcellino, M. (2018). *Using low frequency information for predicting high frequency variables*. International Journal of Forecasting, 34(4), 774-787.

[4] Francq, C., & Zakoïan, J. M. (2010). *GARCH Models: Structure, Statistical Inference and Financial Applications*. Wiley.

[5] García, M. C., Jalal, A. M., Garzón, L. A., & López, J. M. (2013). *Métodos para predecir índices bursátiles*. Ecos de Economía, 17(37), 51-82.

[6] Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edition. Springer.

[7] Hoseinzade, E., & Haratizadeh, S. (2019). *CNNpred: CNN-based stock market prediction using a diverse set of variables*. Expert Systems with Applications, 129, 273-285.

[8] Huertas, A. (2015). *Modelos predictivos para el mercado FOREX*. Master's thesis, Universidad de Murcia.

[9] Hyndman, R. J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*. 3rd edition. OTexts: Melbourne, Australia.
`https://otexts.com/fpp3/` (accessed in August 2023)

[10] Hyndman, R. J. (2012). *Exponential smoothing and regressors*.
`https://robjhyndman.com/hyndsight/ets-regressors/` (accessed in August 2023)

[11] Joseph, M. (2022). *Modern Time Series Forecasting with Python*. Packt.

[12] Kolassa, S. (2017). Answer in *Holt Winters with exogenous regressors in R*. Version of 2017-04-13.
`https://stats.stackexchange.com/q/220885` (accessed in August 2023)

[13] Li, X., Li, Y., Yang, H., Yang, L., & Liu, X. Y. (2019). *DP-LSTM: Differential privacy-inspired LSTM for stock prediction using financial news*. arXiv:1912.10806 [q-fin.ST]

[14] Lones, M. A. (2023). *How to avoid machine learning pitfalls: a guide for academic researchers*. arXiv:2108.02497v3 [cs.LG]

[15] Osman, A. F., & King, M. L. (2015). *A new approach to forecasting based on exponential smoothing with independent regressors*. Monash Econometrics & Business Statistics Working Papers.

[16] Pan, Y., Xiao, Z., Wang, X., & Yang, D. (2019). *A multi-indicator multi-output mixed frequency sampling approach for stock index forecasting.* Romanian Journal of Economic Forecasting, 22(4), 100.

[17] Peixeiro, M. (2022). *Time Series Forecasting in Python.* Manning.

[18] Sax, C., & Steiner, P. (2013). *Temporal disaggregation of time series.* The R Journal, Volume 5/2.

[19] Seth, A. (2007). *Granger causality.* Scholarpedia, 2(7):1667.

[20] Stock, J. H., & Watson, M. W. (2018). *Introduction to Econometrics.* 4th edition. Pearson Series in Economics. Pearson.

[21] Taylor, S. J., & Letham, B. (2017). *Forecasting at scale.* PeerJ Preprints 5:e3190v2

[22] Tsay, R. S. (2010). *Analysis of Financial Time Series.* 3rd edition. Wiley.

[23] Usmani, S., & Shamsi, J. A. (2023). *LSTM based stock prediction using weighted and categorized financial news.* Plos one, 18(3), e0282234.

[24] Wu, H., Xu, J., Wang, J., & Long, M. (2021). *Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting.* Advances in Neural Information Processing Systems, 34, 22419-22430.

[25] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). *Informer: Beyond efficient transformer for long sequence time-series forecasting.* In Proceedings of the AAAI conference on artificial intelligence, Vol. 35, No. 12, pp. 11106-11115.

[26] Aroussi, R. *Library yfinance: Download market data from Yahoo! Finance's API.* `https://github.com/ranaroussi/yfinance`

[27] *Apple Google Trends Data.* `https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0k8z&hl=en`

[28] *Implementation of an ETS model in Python with statsmodels.* `https://www.statsmodels.org/dev/generated/statsmodels.tsa.exponential_smoothing.ets.ETSModel.html`

[29] *Implementation of a Prophet model in Python.* `https://facebook.github.io/prophet/docs/quick_start.html`

[30] *Implementation of a SARIMA model in Python with pmdarima.* `https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html`