Universidad Pontificia de Comillas - ICADE

FORECASTING OF FINANCIAL TIME SERIES UTILIZING GAUSSIAN PROCESS MODELS

Student Key: 201702748

Author: Jaime Reglero García

Director: María Coronado Vaca

MADRID | June 2022

Resumen

El análisis y estudio de series financieras con fines predictivos es uno de los problemas de mayor, y con mayor variedad de enfoques. Como tal, dicho campo de estudio está en constante crecimiento, y la manera de abordar el problema supone un proceso de evolución continua. Una manera de abordar dicho problema es la aplicación de modelos basados en procesos Gaussianos, cuya aplicación al mundo financiero no se ha extendido en gran medida.

Los modelos basados en procesos Gaussianos representan un ejemplo particular de modelos Bayesianos no paramétricos. Esencialmente, se puede reducir un proceso Gaussiano a una distribución de variables que puede ser definida mediante a una media y una matriz de covarianza N x N, siendo N el número de variables de la distribución. Los modelos Bayesianos, además, permiten disminuir la incertidumbre de una variable realizando observaciones de otra variable. Esto proporciona a los modelos basados en procesos Gaussianos una capacidad de adaptación y robustez única, que permite a los modelos adecuarse a los datos en mayor medida.

Este trabajo presenta una primera aproximación al uso de modelos basados en procesos Gaussianos, buscando estudiar no solo sus capacidades predictivas sino también su versatilidad y capacidad de adaptación a series financieras de diferente naturaleza. Como tal, se realiza también un estudio sectorial de las diferentes empresas cotizadas, buscando evaluar los modelos en empresas de todos los sectores de la economía española.

Palabras clave: Modelos basados en procesos Gaussianos, Machine Learning, predicciones financieras, IBEX-35, modelos Bayesianos

Abstract

The analysis and study of financial series for forecasting purposes is one of the largest and most varied of problems. As such, this field of study is constantly growing, and the way of approaching the problem is a process of continuous evolution. One approach to the problem is the application of Gaussian process models, which have not been widely applied to the financial world.

Gaussian process models represent a particular example of non-parametric Bayesian models. Essentially, a Gaussian process can be reduced to a distribution of variables that can be defined by a mean and an N x N covariance matrix, where N is the number of variables in the distribution. Bayesian models, in addition, allow the uncertainty of one variable to be reduced by making observations of another variable. This provides models based on Gaussian processes with a unique adaptability and robustness, which allows the models to fit the data to a greater extent.

This paper presents a first approach to the use of models based on Gaussian processes, seeking to study not only their predictive capabilities but also their versatility and adaptability to financial series of different nature. As such, a sectoral study of the different listed companies is also carried out, seeking to evaluate the models in companies from all sectors of the Spanish economy.

Keywords: Gaussian process models, Machine Learning, financial predictions, IBEX-35, Bayesian models

INDEX

FIGURE INDEX

TABLE INDEX

LIST OF ABBREVIATIONS

BME	Bolsas y Mercados Españoles
Bn	Billion
CAGR	Compound Annual Growth Rate
ETF	Exchange Traded Fund
GP	Gaussian Process
LSTM	Long Short-Term Memory
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
Tn	Trillion
UST	TerraUSD

Introduction

Financial markets are some of the most crucial entities in the world, being responsible for the efficient allocation of assets and capital and distributing risk amongst the different parties involved in the markets. Further, in terms of size, they have shown constant growth over the past decade, particularly in terms of global equity markets capitalization value, reaching a value of $105.8 Tn, and exhibiting a CAGR of 6,2% throughout the years ranging from 2010 to 2020 (SIFMA, 2021).

Given the immense size and importance financial markets carry in the performance and operation of the world´s economy, they are subject to numerous studies regarding their future performance. As such, many different methods have been used to try to predict the different indicators regarding the markets, whether that is a certain stock or index price, or the yield of a country´s bond.

Nonetheless, recent world events have shown that certain one-off events carry great volatility and unpredictability, such as the events regarding GameStop's stock price, which increased close to 30x due to a retail investor frenzy caused by Wall Street Bets, a subreddit from the popular social media platform Reddit (Sánchez, 2021). This particular event, in which a short squeeze was driven by the previously mentioned platform, caused short sellers estimated losses of around $6 Bn, and forced numerous hedge funds to close down, as a consequence of the incurred losses. Other markets, such as the cryptocurrency market, have suffered similar events, such as the recent crash of the Luna cryptocurrency and UST, its associated stable coin. Stable coins are cryptocurrency coins intended to remain stable, acting as a safeguard for investors, and allowing them to retain their capital stably. In this case, UST was designed to always retain a value of $1, but its price dropped as much as 10 cents on May 7th. All in all, Luna and UST´s crash incurred a loss of close to $17 Bn, which no one was able to predict (Van Boom, 2022).

The great importance that financial markets carry, as well as their volatility and unpredictability, have led investors to employ a great variety of methods in the prediction of market performance, particularly the global stock market. Historically, stock analysis methods have been categorized into two precise methods: fundamental analysis and technical analysis (Corporate Finance Institute, 2022).

Fundamental analysis, as its name suggests, involves evaluating a company based on its business fundamentals to predict the intrinsic value of a stock. In doing so, it analyzes the various financial statements a company posts and combines them with various environmental factors such as the industry´s performance and outlook, and the macro-economic outlook. As such, fundamental analysis requires investors to understand not only the company´s business model and operations but also the various external factors that may affect its future value. Notable proponents of this strategy include Benjamin Graham and Warren Buffett, the most notorious advocates of the value investing philosophy.

On the other hand, technical analysis seeks to identify trends and patterns of different stocks, mainly focusing on market data such as stock price and trade volume. Technical analysts, therefore, believe historical price patterns and trading activity can determine a stock´s future price. In doing so, investors that employ technical analysis believe that past historical trends are an indicator of the future trend a stock will show. Additionally, they believe the market takes a company´s fundamentals into account and factors such as information in pricing a stock. Because of this, a company´s fundamentals seem irrelevant to them, given that such information is already taken into account by the current stock price. Most notably, technical analysis theory was developed by Charles H. Dow in a series of editorials in the Wall Street Journal (Nti, Adekoya & Weyori, 2020).

In recent years, Machine Learning and Artificial Intelligence have grown exponentially, with their use increasing across all fields, including finance. As such, modern stock analysis has employed the use of Machine Learning models to forecast the future performance of stocks. According to Shah et al. (2019), these modern techniques can be classified into five different categories: statistical, pattern recognition, machine learning, sentiment analysis, and hybrid. Figure 1 shows a further, more detailed classification, as proposed by Shah et al. (2019).

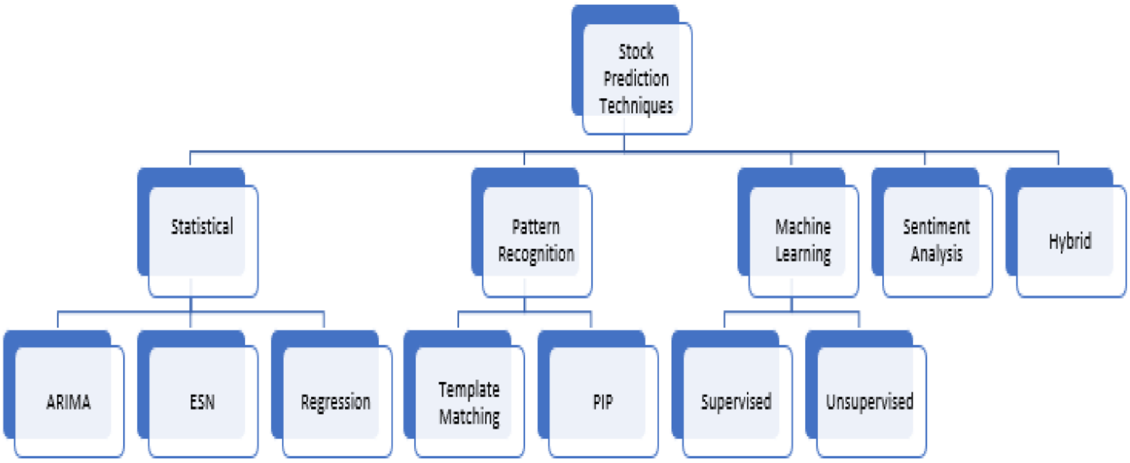

Figure 1: Modern Stock Prediction Techniques

[Source: (Shah, Isah & Zulkernine, 2019)]

As we can see in Figure 1, advances in machine learning techniques, and statistical computing, have led to a wide variety of stock prediction techniques being employed, most notably ARIMA models, and machine learning techniques such as Random Forest regression or Neural Networks.

Throughout this document, we will explore the use of Gaussian processes in the forecasting of financial time series, as well as their versatility when used with different types of stocks. Gaussian processes constitute a framework of Bayesian modelling and represent a versatile method to perform robust predictions in uncertain situations, by using Bayesian inference (Roberts, Osborne & Co, 2013).

Objectives

As we have previously mentioned, financial time series are a constant object of study, and the possibility of forecasting them is of great interest. In this case, one of the main objectives of this study is to assess the forecasting capacity of Gaussian process models, particularly when applied to financial time series. Furthermore, it is of great interest to evaluate the versatility of such models when applied to different financial time series, and, in more detail, whether Gaussian process models are better suited to certain time series. In that sense, we will thoroughly examine the application of Gaussian process models to a wide variety of financial sectors, studying which sector´s companies are more suitable for using Gaussian process models.

Methodology

The purpose of this dissertation is to assess and evaluate the accuracy and versatility of Gaussian process models in the forecasting of financial time series. As such, a valid and extensive process must be followed.

The first phase of the project will be to study and research the available literature regarding Gaussian processes. Gaussian process models are mathematically complex and complicated models, and, as such, they require an initial period of observation and exploration to properly grasp their conceptual and fundamental basis. As part of this phase, we will also explore the available literature regarding the current uses of Gaussian process models, as well as their limitations. This phase will provide us with a solid foundation that will enable the correct use and application of Gaussian process models.

The second phase will be a study and analysis of the different financial time series. Given the nature of the financial markets, narrowing down the scope of analysis will be crucial, to facilitate the application of the Gaussian process models. As such, we will employ the classification used by the Spanish Stock Exchange (BME) and will choose a representative company from each sector to apply the Gaussian process models and forecast such companies.

Having understood the basis of Gaussian processes and analyzed and reduced the available scope of financial time series, we will then apply the Gaussian process models. Such application will be performed on Python, with the target variable we want to predict being the monthly returns. As such, the models will receive the monthly stock price and predict the following month´s return, based on a training and testing dataset.

Finally, as with all studies, it will be of the utmost importance to perform an extensive and exhaustive analysis of the results, to extract valuable information and useful conclusions. Such conclusions will revolve around the predictive capabilities of Gaussian process models, and their versatility when applied to a great variety of financial time series.

Chapter 1: Gaussian Processes

In this chapter, we will analyze the conceptual and mathematical foundations of Gaussian processes, as well as their current uses, and their possible applications to financial time series of various kinds.

Firstly, though, we must establish the problem at hand. Throughout this document, we will try to predict and model a financial time series. Therefore, the problem at hand can be classified as supervised learning, a subcategory of machine learning in which an input is mapped to an output. More specifically, given that the desired output is a numerical one, the problem will be classified as a regression problem.

Within regression problems, there are numerous different approaches one can take to solve the problem, ranging from simple methods such as linear regression to more complex methods such as neural networks. As such, one of the many options is to assign probabilities to the range of possible functions, with those functions that are more likely to have a higher probability (Rasmussen & Williams, 2005). In this case, we will use Gaussian processes as a generalization of the Gaussian probability distribution. However, to fully understand the concept of Gaussian processes and their application to financial time series, we must first delve into the concept of Bayesian time series analysis and modeling.

1.1 Bayesian Modeling

Bayesian models employ probabilities to represent uncertainty within the model. As such, they are one of the many types of non-parametric models, that is models that assume that a certain set of data points cannot be defined in terms of a set of finite parameters. Examples of parametric models include linear regression problems, in which a set of coefficients is adjusted to fit a certain set of data points. Non-parametric models, as a result, have greater freedom to adjust the model to the given data points, and, in some cases can display a much more robust performance. The difference between a parametric model, ordinary least squares, and a non-parametric model, is shown below (Gurudath, 2022):

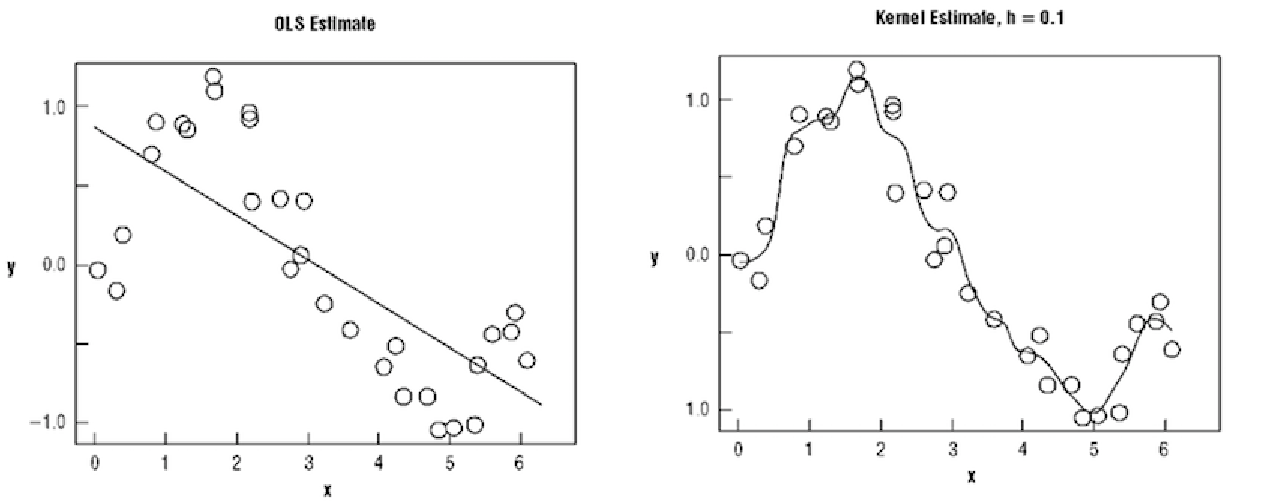

Figure 2: Parametric Model vs Non-Parametric Model

[Source: (Gurudath, 2022)]

Bayesian models, such as the one shown in Figure 2, treat the learning process as an inference problem. In this sense, the inference problem is based on a set of observations or data points, and a set of curves is fit to such a set of data points. As such, this allows us to have much higher variability in regions where we have no observations. This is very clearly seen in the following figure displayed by Roberts, Osborne & Co, (2013), in which a set of data samples located at x = 0, 1, 2, and its target values are modelled using a least-squares regression model, and Bayesian modelling:

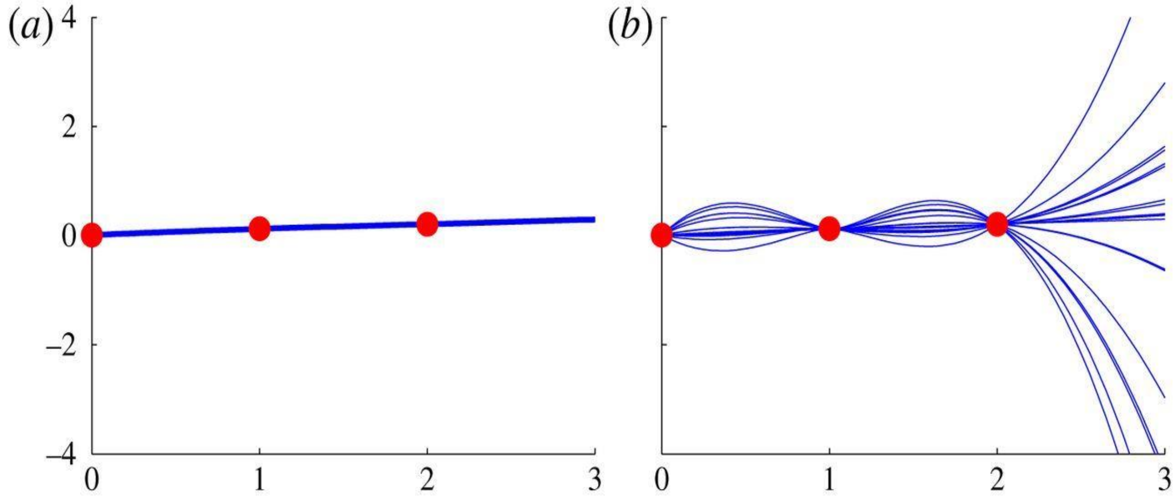

Figure 3: Least-squares model vs Bayesian Modeling

[Source: (Roberts, Osborne & Co, 2013)]

As we can see from Figure 3, while the least-squares model fits the data points perfectly, using Bayesian inference, we can predict with a much higher degree of variability. As such, the chance of fitting our curve to a region without observation of data points is much higher, displaying a much more robust prediction. This particular characteristic is one of the main strengths of Bayesian models, and Gaussian processes in particular, which allows them to fit unpredictable, non-linear data such as financial time series.

1.2 Gaussian Processes

Gaussian processes, as previously mentioned, represent a particular example of Bayesian non-parametric models. According to (Rasmussen & Williams, 2005), a Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. As such, considering a two-variable distribution (x1, x2), we can define a Gaussian distribution by a mean and an N x N covariance matrix, which, in this case, given a two-variable distribution will be 2 x 2. Therefore, by performing Bayesian inference, and providing a relationship between the two variables described by the covariance matrix, an observation of one of the variables allows us to decrease the uncertainty in the other. Figure 3 (Roberts, Osborne & Co, 2013) shows an example of the mentioned concept, in which the mean (black line) and variance (gray shaded region) of x1 and x2 are portrayed before and after observing x1:

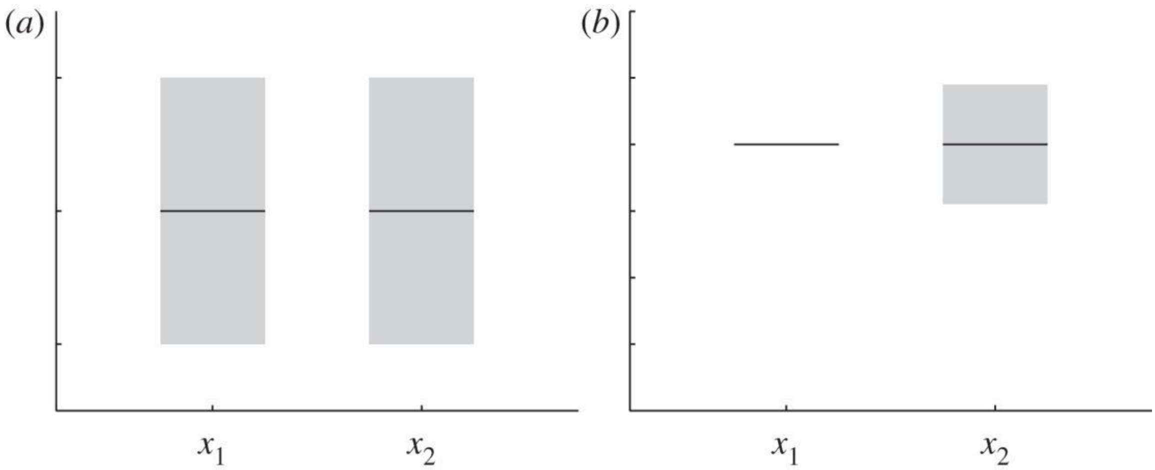

Figure 4: Mean and variance of x1, x2 before and after observation of x1

[Source: (Roberts, Osborne & Co, 2013)]

Figure 3 shows an example of two variables, x1 and x2, to display a simple, understandable example. However, this same concept can be extrapolated to a larger number of variables, which allows us to evaluate a probability distribution of a function at any location, by using a certain number of observed data points.

1.2.1 Covariance Kernel Matrix

As previously mentioned, the covariance matrix represents a key aspect of Gaussian processes, given the impact it has on the uncertainty of variables based on a different variable. In essence, the covariance matrix represents the impact an observation of a certain variable has on the uncertainty of another variable.

The covariance kernel matrix, therefore, defines the covariance of two elements (x1, x2) based on a certain kernel function K(x1, x2). This kernel function, K, along with the mean function defines the Gaussian process distribution and therefore constitutes a crucial element of Gaussian processes.

According to Duvenaud (2014), we can divide the different kernel functions into two types: standard kernels and combining kernels. Standard kernels are traditional kernels that are based on single functions while combining kernels allow us to build a kernel over different types of data by combining different kernels.

1.2.1.1 Standard Kernels

Standard kernels represent the most traditional and simple kernel functions used. Amongst standard kernels, we can name the following:

- Constant Kernel
- White Noise Kernel
- Squared Exponential Kernel (RBF)
- Rational Quadratic Kernel
- Periodic Kernel

The first, and most simple kernel, is the constant kernel. As the name suggests, the kernel function is defined by a constant value k:

$$Kconst(x, x') = k$$

Secondly, we must mention the white noise kernel, which represents uncertainty given by a certain variance σ. Given that white noise aims to represent uncertainty, this kernel is most often used in addition to a different kernel.

$$Kwn(xn, xm) = \sigma * \delta(\mathrm{n}, \mathrm{m})$$

The squared exponential kernel, also known as the radial basis function, is represented by the following equation, where h is an output amplitude, and l is a time scale.

$$Kse(x, x') = \mathrm{h}^2 * \exp\left(-\frac{(x - x')^2}{2l^2}\right)$$

Figure 5, courtesy of Natsume (2022) depicts the squared exponential kernel graphically:

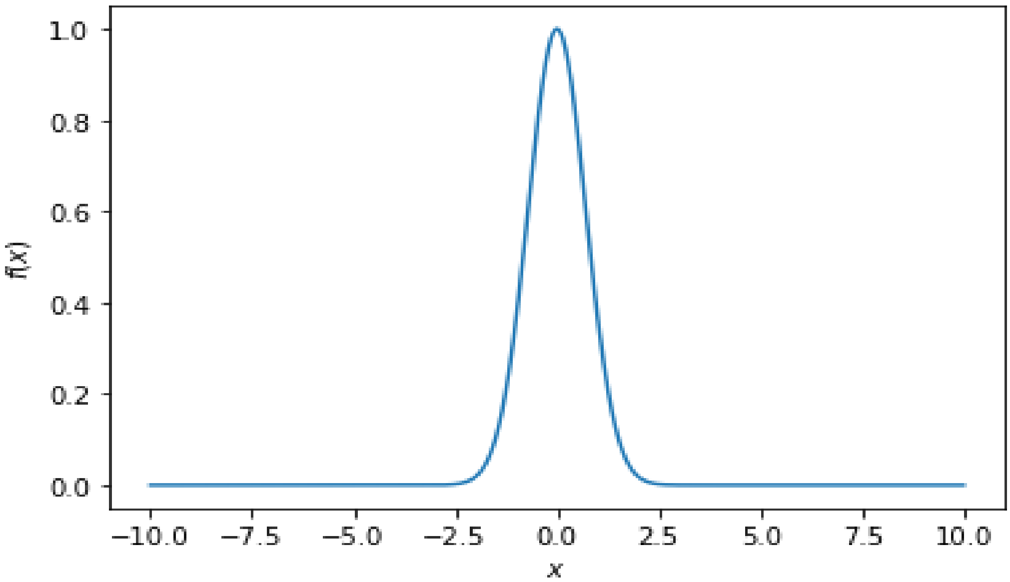

Figure 5: Squared Exponential Kernel

[Source: (Natsume, 2022)]

On the other hand, the rational quadratic kernel is given by the following equation, where α is the index, which determines the weight of length scale variations:

$$Krq(x, x') = \mathrm{h}^2 * \left(1 + \frac{(x - x')^2}{\alpha * \mathrm{l}^2}\right)^{-\alpha}$$

Figure 6, courtesy of Natsume (2022) depicts the rational quadratic kernel graphically:

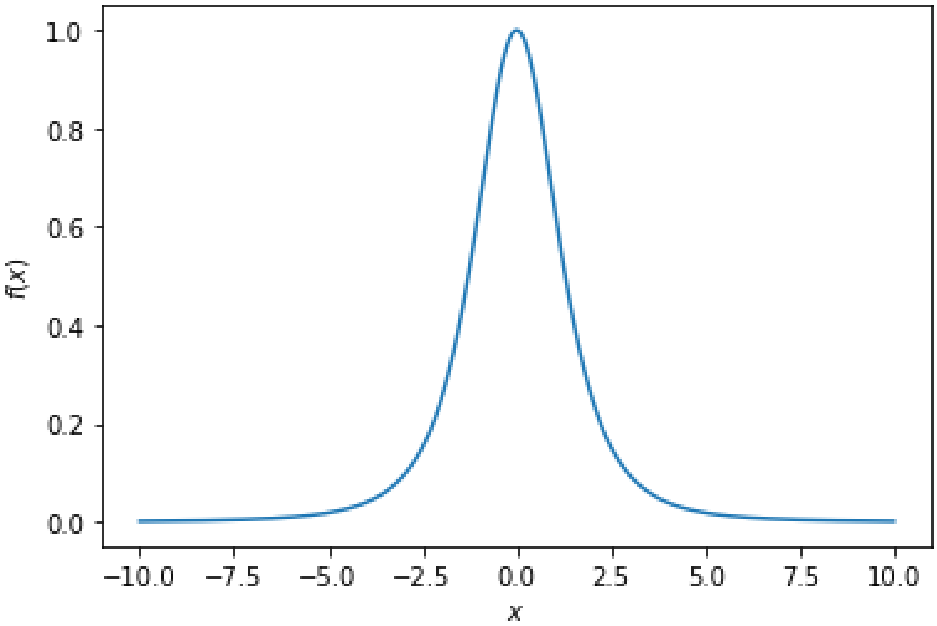

Figure 6: Rational Quadratic Kernel

[Source: (Natsume, 2022)]

Furthermore, the rational quadratic kernel is the equivalent of adding many different squared exponential kernels with varying length scales. Additionally, it is worth mentioning that the rational quadratic and squared exponential kernels are the kernels most commonly used, due to their versatility in fitting to functions with no discontinuities.

Finally, the periodic kernel, also named the exponential sine squared kernel, models periodic functions, and is given by the following equation, with p representing the period of the function, and l the length scale:

$$Kper(x, x') = \mathrm{h}^2 * \exp\left(-\frac{2\sin^2\left(\pi * \frac{|x - x'|}{p}\right)}{l^2}\right)$$

Figure 7, courtesy of Natsume (2022) depicts the periodic kernel graphically:

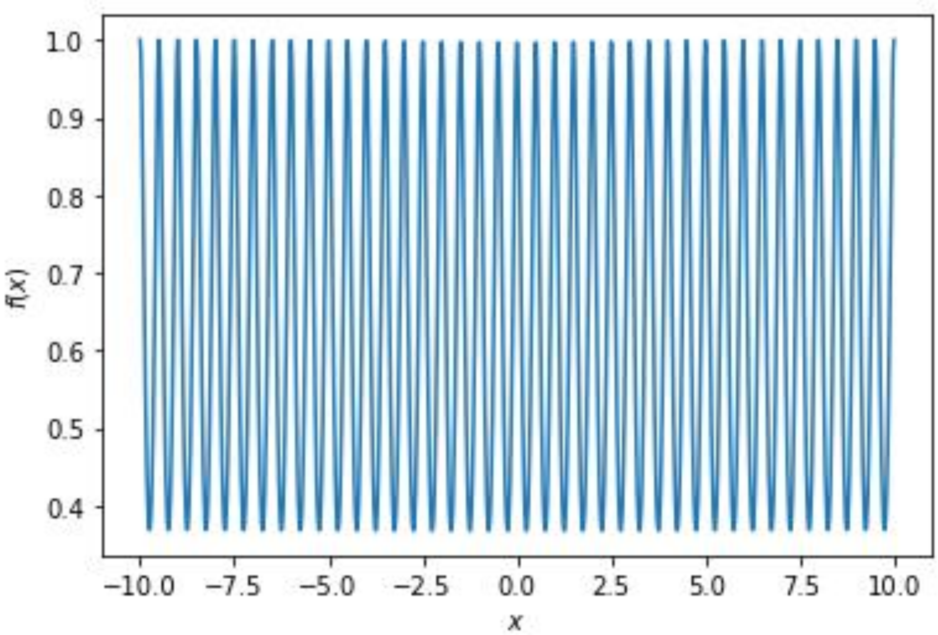

Figure 7: Periodic Kernel

[Source: (Natsume, 2022)]

1.2.1.2 Combining Kernels

Previously, we mentioned and delved into the different standard kernel functions available, or at least those most used. However, often standard kernels are not optimal, and some occasions require different types of kernels. On these occasions, combining kernels allows one to develop a more complex kernel function that fits itself in a better way to the data available. Amongst combining kernels there are two main options available:

- Additive Kernels
- Multiplying Kernels

As the name suggests, additive kernels combine two kernels by adding them together. Popular combinations of such kernels include the sum of an RBF or squared exponential kernel and white noise, to capture a long-term rising trend, while also taking uncertainty and noise into account.

On the other hand, multiplying kernels result in the multiplication of the kernel functions being multiplied. Conceptually, this would mean that the resulting kernel would have a high value only if both kernels have a high value. The most popular combinations of multiplying kernels are a periodic kernel and a squared exponential kernel, which results in a locally repeating function, that is, a function that does not repeat itself exactly.

1.2.2 Mean Functions

As previously mentioned, a Gaussian process distribution is defined by a mean function and a covariance kernel matrix that defines the probability relationship between variables. Having delved into the different covariance kernels, we will now briefly introduce the concept of mean functions.

Most commonly, Gaussian processes are initially considered to have a mean function of zero. While this might not seem logical, when inference occurs this mean function value will change, which is the reason why the initial mean function is not of the utmost importance. If, however, one knows of the magnitude or trend of the mean function, such a mean function could be incorporated into the Gaussian process.

1.2.3 Hyperparameters

Previously, we briefly delved into the concept of non-parametric models compared to parametric models, asserting that Gaussian processes are based on Bayesian non-parametric models. Nonetheless, Gaussian process models have certain hyperparameters, derived from the different covariance functions mentioned above. For example, in the case of the squared exponential kernel, or radial basis function, the length scale (l) and signal variance (h) can both be varied to fit the kernel. These free parameters are what we will denote as hyperparameters.

As is expected, these hyperparameters have a big impact on the outcome of the model, and it will be very important to properly finetune these hyperparameters to ensure proper model fit, and, ultimately, the optimal prediction interval. While conceptually, this might seem difficult to understand, graphical examples portray this very well. For example, Figure 8 (Rasmussen & Williams, 2006) portrays the difference between an optimal length scale (l = 1), a too-short length scale (l = 0.3), and a too-large length scale (l = 3), particularly regarding the amplitude of the 95% confidence interval of the predicted output:

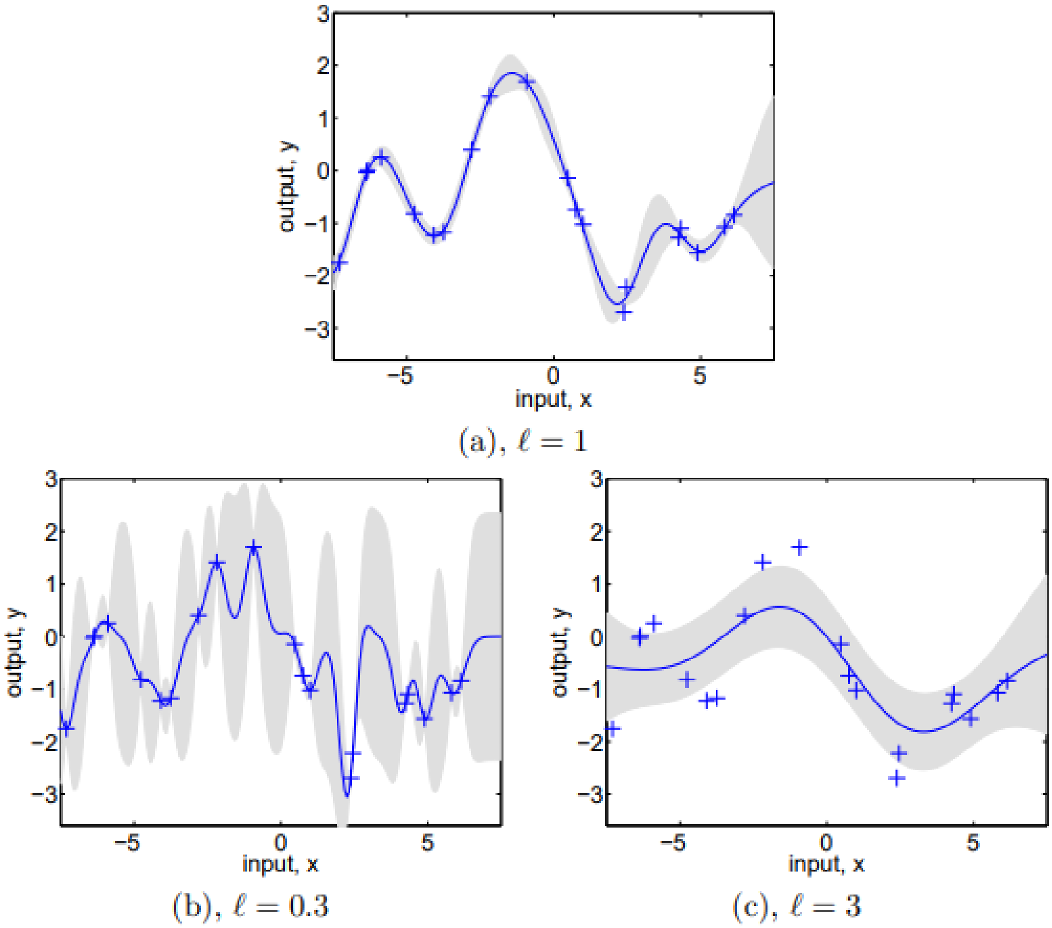

Figure 8: Gaussian Process Prediction with varying length scales

[Source: (Rasmussen & Williams, 2006)]

As we can see, the tuning of hyperparameters can become a crucial aspect of the model selection and training, and, as such, understanding the available data points, and how different kernel functions will adapt to such data will be of the utmost importance.

Chapter 2: Financial Time series Analysis

Having established the conceptual framework of Gaussian processes and the different components and aspects related to GP models, we will now delve into the different financial time series we will analyze and forecast.

2.1 Spanish Stock Exchange (BME)

Firstly, we must develop an adequate scope of the vast quantity and variety of different financial time series at hand. As such, throughout this document we will focus on the equity markets, with a particular focus on the Spanish equity market, and, as such, will analyze some of the most representative companies in the Spanish stock market. Given that the Spanish equity market has close to 2600 listed companies (BME, 2022), an adequate classification and taxonomy should be established to get an ample and extensive analysis of the market. As such, we will follow the sectoral classification established by the Spanish Stock Exchange (BME), which distinguishes the following sectors:

- Energy and Oil
- Materials, Industry and Construction
- Consumer Goods
- Consumer Services
- Financial Services
- Technology and Telecommunications
- Real Estate Services

In the following chapter, we will briefly delve into the recently mentioned sectors, and discuss several of the most notable companies which we will analyze throughout this document.

2.1.1 Energy and Oil

The first category of companies listed on the Spanish Stock Exchange is the energy and oil category. Companies under this umbrella include companies engaged in the exploration, extraction, production, and refining of oil products, as well as gas and electricity distributors. As such, BME distinguishes four sub-categories within the Energy and Oil category:

- Oil
- Electricity and Gas
- Water and Others
- Renewable Energies

The oil sub-category encompasses companies involved dedicated to oil products and services related to oil and oil derivatives, most notably Repsol. Electricity and gas, on the other hand, is comprised of companies whose main commercial activity is the production or distribution of gas and electricity, with companies such as Iberdrola, Endesa or Naturgy included under this sub-sector. The Water and Others sub-category includes companies whose main activity is the supply of water or other activities such as the treatment of nuclear or radioactive waste. Finally, the Renewable Energies sub-sector consists of companies whose main activities include the production of electricity from renewable sources, which includes companies such as Acciona Renovables.

Given the wide range of activities encompassed under the Oil and Energy category, analyzing a company with exposure to the various sub-sectors recently explained will be of great importance. As such, we will further analyze and forecast the stock price of Iberdrola, given their ample exposure to renewable energies, their established position as leading electricity providers, and their presence in all the major global geographical markets.

2.1.2 Materials, Industry and Construction

The second category of companies is the Materials, Industry and Construction category, which encompasses companies dedicated to the extraction and treatment of raw materials, general construction activities, or various industries such as the chemical, engineering, and aerospace industries. Within such category, we can distinguish the following sub-sectors or sub-categories:

- Raw Materials
- Manufacturing and Assembly
- Construction

- Construction Materials
- Chemical Industry
- Engineering and Others
- Aerospace

The first sub-sector we must discuss is the Raw Materials sub-category. This sub-category includes companies dedicated to the exploration, extraction, and refining of raw materials and manufacture of alloy products, with the most notable companies being Acerinox and ArcelorMittal. The Manufacturing and Assembly sub-sector, on the other hand, includes companies whose primary activities revolve around the manufacture and assembly of machinery and equipment of various types, with the most notorious companies being Gestamp and Talgo. The construction sub-category, as its name states, includes companies whose main commercial activity revolves around the construction of civil or military works, with companies such as ACS or Ferrovial englobed under this sub-sector. Following on, the Construction Materials sub-sector includes companies whose principal activity is the manufacture and extraction of non-metal construction materials. The Chemical Industry sub-sector, on the other hand, focuses on companies whose main activity is the manufacture of basic chemicals and chemical products. The Engineering and Others sub-category includes companies dedicated to performing civil or military engineering activities, with Abengoa and Tecnicas Reunidas being the principal companies. Finally, the Aerospace sub-sector covers companies involved in the manufacture or distribution of airplanes and aircraft components.

As we can see, Materials, Industry and Construction cover a wide range of activities, and, as such, the chosen company to analyze must be one with exposure to a wide variety of industries. With that in mind, we will choose ACS as the company to forecast, given its ample exposure to various industries and geographic markets.

2.1.3 Consumer Goods

The third category of companies is the Consumer Goods category, which covers companies dedicated to commercial activities such as the production and distribution of food and beverage, manufacturing, distribution, and sale of textile products, as well as companies involved in the development of pharmaceutical and biological products. Furthermore, we can distinguish the following sub-sectors:

- Food and Beverage
- Textile Products
- Paper and Graphic Arts
- Automobile
- Pharmaceutical and Biotech Products
- Other Consumer Goods

Firstly, the Food and Beverage sub-sector focuses on companies whose main economic activity is the production, processing, or marketing of food products and beverages, as well as companies dedicated to the maintenance of crops and livestock. Notable companies included under this sub-sector are Pescanova or Coca Cola Europacific Partners. The Textile sub-sector includes companies such as Inditex, that is, companies whose main activity revolves around the production, distribution, and sale of textile products and clothing. The Paper and Graphic Arts sub-category involves companies that, as its name suggests, focus on the production and treatment of paper and cardboard products. Meanwhile, the Automobile sub-sector covers companies that focus on manufacturing automobiles, motorcycles, and other vehicles. On the other hand, the Pharmaceutical and Biotech Products sub-sector includes companies dedicated to the manufacture and distribution of medicines and vaccines, as well as those involved in the research and development of biological products, with companies such as Grifols or Armirall englobed under this sub-category. Finally, companies whose main commercial activity is unfit for other categories fall under the Other Consumer Goods sub-sector.

In this case, given the nature of consumer goods, and the many different forms of such a product, we have chosen to forecast and analyze Inditex, given their international exposure and recognition, and the importance of the textile industry within consumer goods.

2.1.4 Consumer Services

The fourth category the Spanish Stock Exchange uses to classify companies is Consumer Services, which includes companies engaged in leisure and tourism activities, commerce, media or advertising, and transport services. Additionally, the following sub-categories are identified:

- Leisure, Tourism and Hotels
- Commerce
- Media and Advertising
- Transport and Distribution
- Motorways and Parking
- Other

The first sub-sector within the Consumer Services category is Leisure, Tourism, and Hotels, which includes companies engaged in the management of leisure activities and facilities, as well as hotel management companies, with the most notable ones being Melia Hoteles and NH Hotel. The second sub-sector, Commerce, includes companies that focus on activities related to the storage and retail distribution of products to supermarkets and other stores. Thirdly, the Media and Advertising sub-sector encompasses companies whose main activities are related to media, whether that´s radio or television, as well as advertising activities, and includes companies such as Vocento or Atresmedia. The Transport and Distribution sub-sector covers companies that provide transport services to goods or people, as well as those companies dedicated to the packaging, storage, and distribution of goods. This sub-sector includes companies such as IAG, an airline holding group, or AENA, the Spanish airport manager. The Motorways and Parking sub-sector is comprised of motorway concession companies and public parking operators. Finally, other companies such as Prosegur, a security and protection company, or Clinica Baviera, a dental healthcare provider company, are included in the Other sub-category, which fits those companies that don't fit under any of the other sub-categories.

Once again, given the variety of services included within the Consumer Services category, we will analyze and forecast AENA, given that their impact and reach can be traced to a lot of the sub-sectors, and the impact they suffered throughout the COVID-19 pandemic, given that their business relies on airports and air travel.

2.1.5 Financial Services

The fifth category under which the Spanish Stock Exchange classifies companies is Financial Services. As the name suggests, this category includes companies engaged in the banking and insurance industry, as well as investment companies whether that is real estate or securities. Furthermore, the following sub-sectors are distinguished:

- Banks
- Insurance
- Holding Companies
- Investment Companies
- Investment Services
- Traded Funds
- Private Equity
- Hedge Funds

The first sub-sector within the Financial Services industry includes banks and savings banks, companies engaged in the traditional banking activity performed by credit institutions, and includes companies such as BBVA or Banco Santander. The second sub-sector is Insurance, which comprises those companies involved in the commercialization and sale of insurance policies, such as Mapfre or Grupo Catalana Occidente. The third sub-sector, holding companies, includes those real estate investment companies not regulated by any specific law, such as Alantra or Corporacion Financiera Alba. The fourth sub-sector is Investment Companies, otherwise known as SICAV, which encompasses financial instruments whose purpose is to raise, manage and invest funds. The fifth sub-sector, as its name suggests includes companies dedicated to the provision of investment services. On the other hand, the sixth sub-category is Traded Funds, which, as its name suggests includes traded investment funds, including exchange-traded funds (ETF). Finally, the last two sub-sectors are Private Equity and Hedge Funds, which, as their names suggest, focus on venture capital and private equity funds on one hand, and hedge funds on the other.

Within the financial services industry, there is a great variety of sub-sectors, however, most of those sub-sectors are greatly related. As such, a global bank such as BBVA has exposure to many of those sub-sectors, from commercial and investment banking to asset management services, and will, therefore, represent the financial services category optimally.

2.1.6. Technology and Telecommunications

The sixth category specified by the Spanish Stock Exchange is Technology and Telecommunications. This sector encompasses those companies involved in activities related to telecommunications, such as telephone and communication networks, and electronica and software activities. The following sub-sectors are identified:

- Telecommunications and Others
- Electronics and Software
- Technological Hardware

The first specified sub-sector is Telecommunications and Others, which includes companies dedicated to the provision of telco services, as well as the installation and management of communication infrastructure. Furthermore, any other activity not included in the remaining sub-sectors is included in this sub-sector. The most notable example of a company that fits under this sub-sector is Telefónica, the leading national telephone provider in Spain, as well as other providers such as Cellnex. The second sub-sector specified by the Spanish Stock Exchange is Electronics and Software, which includes companies engaged in R&D, production, and operation of applications, systems, and software. Amongst companies included within the Electronics and Software sub-sector are companies such as Amadeus and Indra, both leading companies in Spain. Finally, the third and final sub-sector is Technological Hardware, which comprises companies involved in the manufacturing and distribution of technological hardware, such as computers, servers, or workstations.

In terms of the Technology and Telecommunications category, the most representative company, due to its involvement in not only the provision of telecommunication services but also the management of network infrastructure, is Telefonica. As such, it will be the chosen company to analyze and forecast.

2.1.7 Real Estate Services

The seventh, and final, category specified by the Spanish Stock Exchange under which companies are classified is Real Estate Services. As its name suggests, this sector includes companies whose main commercial activity is the development, management, or exploitation of real estate. Further in-depth, the Spanish Stock Exchange distinguishes two main sub-sectors:

- Real Estate Agencies and Others

- Listed Public Limited Companies investing in the Real Estate Market (SOCIMI)

The first sub-sector under which companies are classified is Real Estate Agencies and Others, which comprises companies whose main commercial activities include the promotion and rental of real estate, as well as those companies involved in leasing, financing, and any other activity not included in the other sub-sectors. Amongst companies listed under this sub-sector, the most notable companies are Metrovacesa and Aedas Homes. Finally, SOCIMIs are publicly listed companies investing in the real estate market, with specific regulations adhered to them. The most notable of such companies are Merlin Properties and Inmobiliaria Colonial.

Within the Real Estate Services category, there are not many differences in concept, but rather in the regulatory requirements and benefits. As such, the chosen company will be Merlin Properties, the leading SOCIMI in Spain, with exposure to many real estate sub-markets, ranging from office buildings to shopping centers or hotels.

Chapter 3: Application of Gaussian Process Models

Throughout this chapter, we will briefly discuss the practical application of Gaussian process models, assessing the results of each sector we will apply the models. As such, we will divide this section into the seven previously mentioned sectors.

It is important to state that the application and development of the Gaussian process models will be made on Python, using the models developed on Scikit-learn, a free software machine learning library with several different model implementations. Among such implementations, one can build models based on Gaussian processes, by using the Gaussian Process Regressor class. Furthermore, we will use other libraries such as Pandas and Numpy to perform exploratory analysis and preprocessing tasks. Lastly, we will download the financial data employing the Yahoo Financials library, which allows us to extract stock data from Yahoo Finance simply and practically.

3.1. Energy and Oil

As we mentioned throughout Chapter 2, within the Energy and Oil sector we will analyze Iberdrola´s stock performance. It is important to mention that the time interval we will study is between January 2017 to February 2022. Iberdrola´s stock price for such time interval is shown in Figure 9:

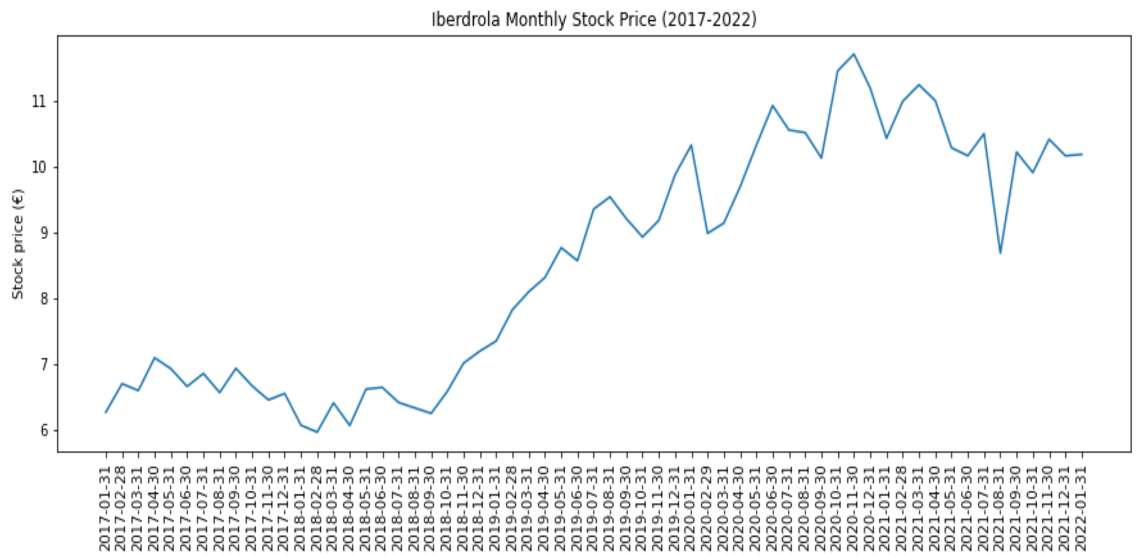

Figure 9: Iberdrola Stock Price

[Source: Own elaboration from Yahoo Finance data]

From Figure 9, we can see that Iberdrola´s stock price has increased steadily, particularly throughout 2018 and 2019, and suffered a slight dip due to the COVID-19 pandemic. We can see Iberdrola´s monthly stock returns in Figure 10:

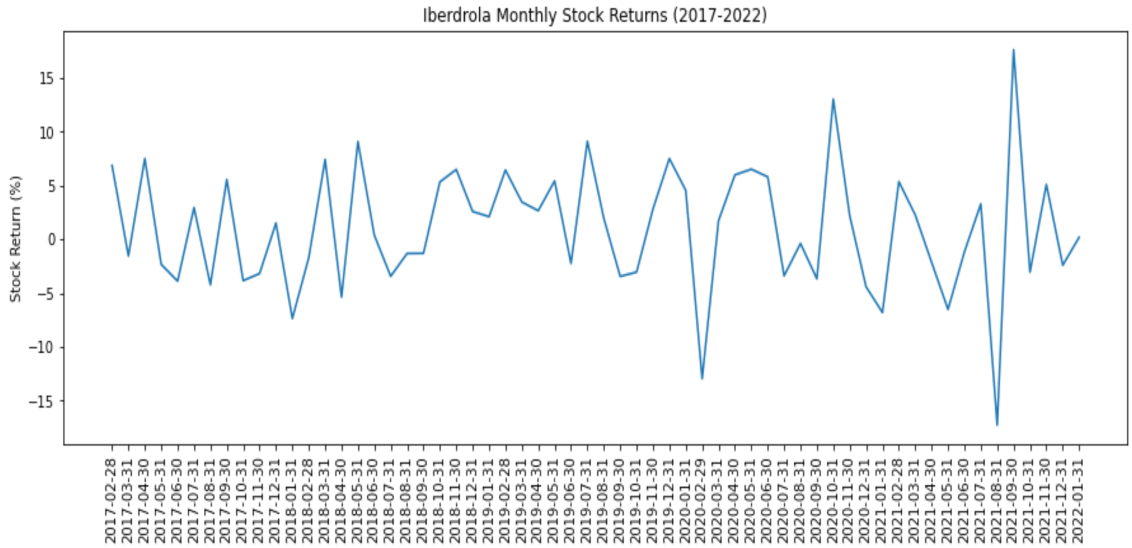

Figure 10: Iberdrola Monthly Stock Returns

[Source: Own elaboration]

As we can see, the monthly returns are much more varied and have suffered ups and downs. In this case, the objective will be to predict the following month´s return, as a percentage, based on the price. To train the model, we will divide the available data into a training and testing dataset. In this case, we have opted for a testing dataset size of 16 observations, equivalent to around 25% of the dataset. Such testing data points will be chosen at random and will be the variable to predict.

The model we will fit will employ a combining kernel that will sum a White Noise kernel with a squared exponential kernel, or radial basis function. Additionally, the model will tune the hyperparameters of each kernel within a specified range, with the squared exponential kernel length scale being bound between 0.00001 and 10000, and the noise level bounds for the white noise kernel between 0.00001 and 10000. Having built the Gaussian process regression model, we then fit the model with our training data, evaluate the testing dataset and obtain a root mean squared error (RMSE) of 3,24 and a coefficient of determination R^2 of 0,9915.

After fitting the model with the training dataset, and evaluating it with the test data, we now predict the entire dataset based on the previously trained model. Figure 11 shows a graphical representation of the monthly stock return prediction and credible interval compared to the real values:

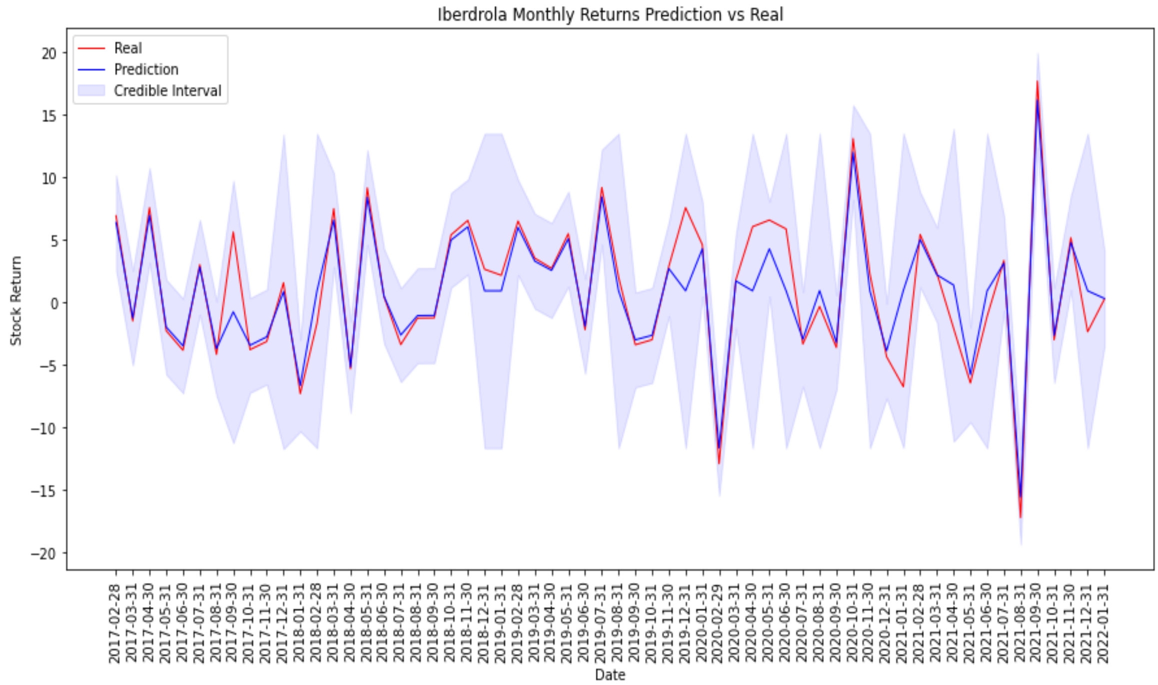

Figure 11: Iberdrola Stock Returns Prediction vs Real

[Source: Own elaboration]

As we can see, the prediction has been very accurate, achieving great performance shown by the coefficient of determination and root mean squared error (RMSE). Furthermore, analyzing the graphical representation we can see that the prediction has been greatly accurate, especially in terms of determining the trend of the stock, whether that´s positive or negative. As such, only on certain occasions does the prediction differ from the real value, and, in such cases, the trend is adequate. The previous analysis and modeling will now be performed for the rest of the pertinent companies.

3.2. Materials, Industry and Construction

After analyzing the energy and oil industry, and, in particular, the application of Gaussian process models to forecast Iberdrola´s stock returns, we will now perform the same task for ACS, the leading international construction company. In Figure 12, we can see the stock price for ACS between January 2017 and February 2022:

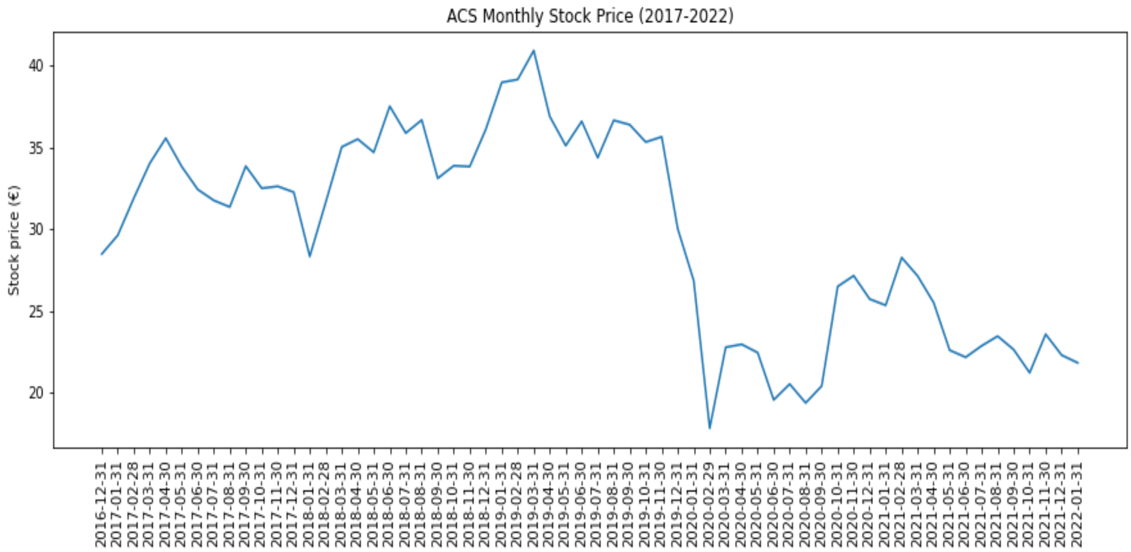

Figure 12: ACS Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

In this case, the price of ACS stock has suffered several drawbacks, particularly in the last quarter of 2019 and the second quarter of 2021. The monthly stock returns are shown in Figure 13:

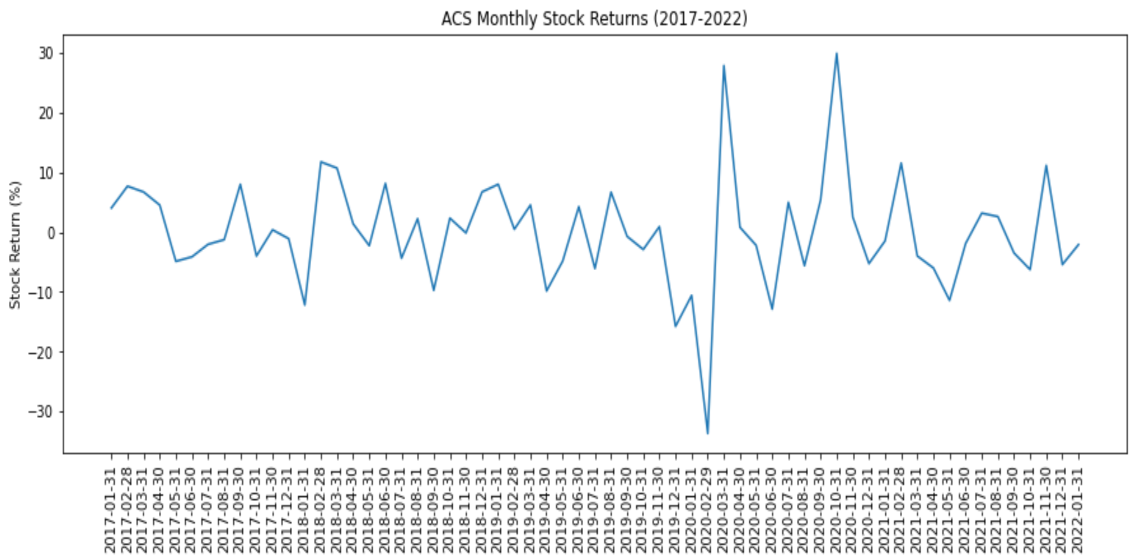

Figure 13: ACS Monthly Returns

[Source: Own elaboration]

As we previously mentioned, the last quarter of 2019 and second quarter of 2021 we particularly bad quarters, and we can see negative returns of up to -20% throughout those periods, as well as a return of -30% when the COVID-19 pandemic impacted the world´s economy. Now, once again, we will fit the model to our training dataset, 75% of the available data, and evaluate its performance on the testing dataset, the remaining 25% of the dataset.

As part of the model, we have opted for a kernel composed of a squared exponential kernel with length scale bounds between 1e-5 and 1e5, and a white noise kernel with noise level bounds between 1e-5 and 1e5. This combination of kernels allows us to model uncertainty using the white noise kernel, while also accounting for the rest of the data with the squared exponential kernel. After fitting the model to our training results, we obtained very strong results, with the RMSE of the testing data being 5,46 and the coefficient of determination, R^2, of 0.992.

After fitting the model with the training dataset, and evaluating it with the test data, we can now predict the entire dataset based on the previously trained model, which will give us a view of the overall predictive capabilities of the model. Figure 14 shows the monthly stock return prediction and confidence interval for each prediction compared to the real values:

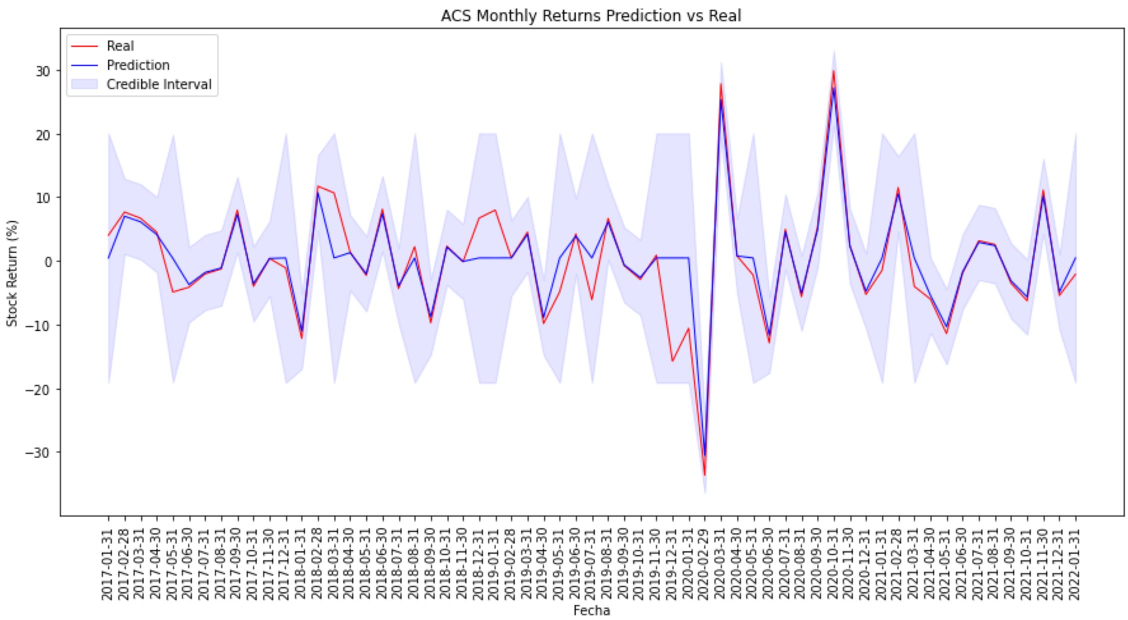

Figure 14: ACS Monthly Returns Prediction vs Real

[Source: Own elaboration]

As we can see, the prediction is very accurate, and only displays certain inaccuracy during the last semester of 2019. Otherwise, certain periods, such as the years 2020 and 2021 show great accuracy and performance when comparing the predictions to the real values.

3.3. Consumer Goods

Now, we will move on to analyze the Consumer Goods industry, specifically Inditex´s stock returns. In Figure 15, we can see Inditex´s stock price between June 2017 and February 2022:

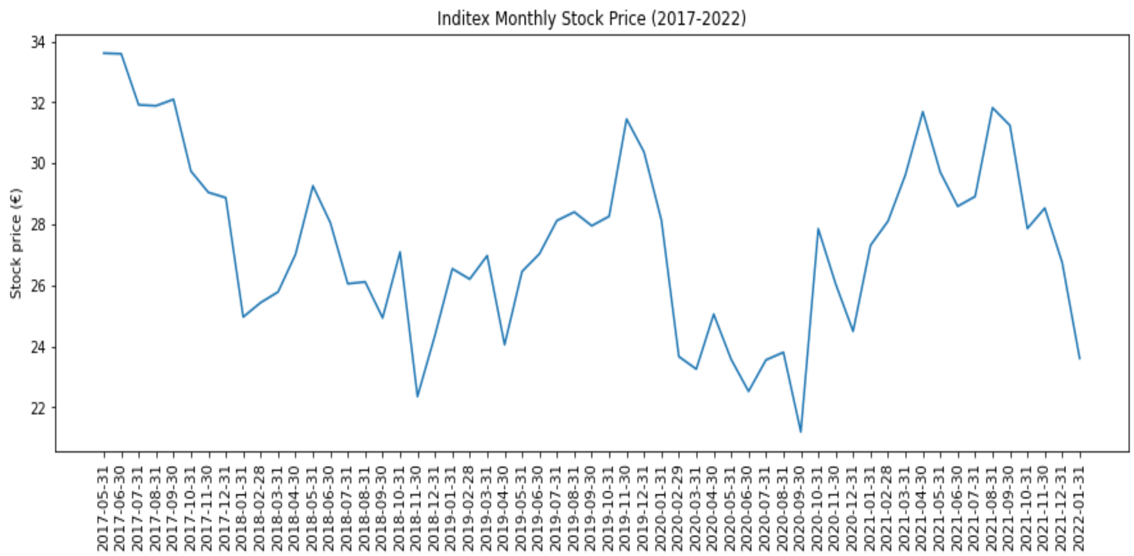

Figure 15: Inditex Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

In this case, the price of Inditex stock had steadily decreased in the past years, particularly throughout the second semester of 2017 when the stock dropped from 34€ to 24€, close to a 30% decrease. However, the price increased throughout 2019, before decreasing again in 2020 due to the COVID-19 pandemic. In Figure 16 we can see Inditex´s monthly returns:

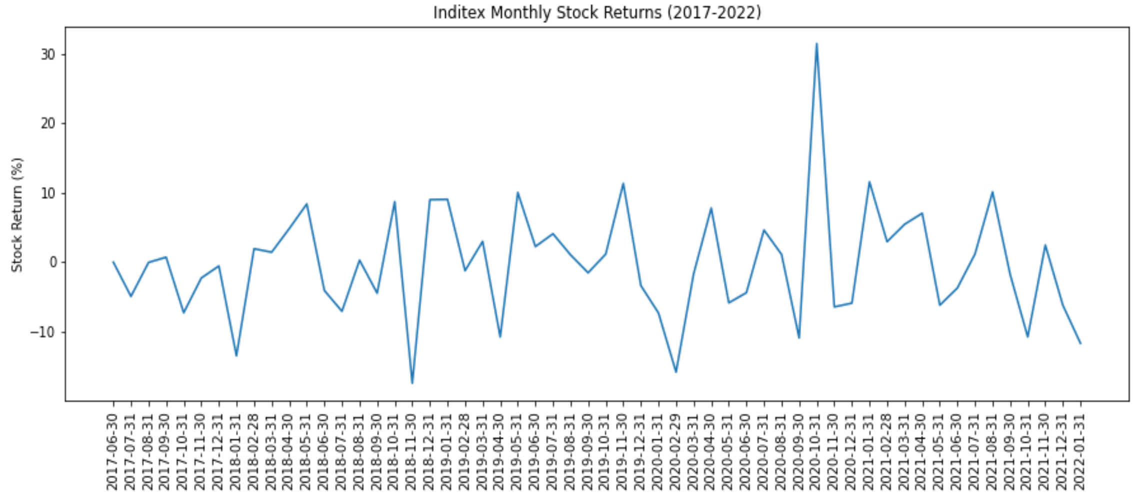

Figure 16: Inditex Monthly Stock Returns

[Source: Own elaboration]

From Figure 16, we can see that Inditex´s stock had particularly bad returns throughout the years 2017 and 2018, as well as the early times of the COVID-19 pandemic.

Concerning the model, we have opted for a kernel composed of a squared exponential kernel with length scale bounds between 1e-7 and 1e6, and a white noise kernel with noise level bounds between 1e-8 and 1e6, which provides the perfect combination to model a volatile time series with uncertainty. After fitting the model to our training results, we obtain an RMSE of the testing data of 6,85 and a coefficient of determination, R^2 of 0.999. As we can see, in this case, the RMSE is slightly higher than in other cases, but the coefficient of determination is higher, which leads us to feel that, in general, the model fits very well, but it deals worse with outliers.

After fitting the model with the training dataset, we now use the trained model to predict the entire dataset, to get an overall idea of the predictive capacity of the model. Figure 17 shows the monthly stock return prediction and confidence interval compared to the real values:

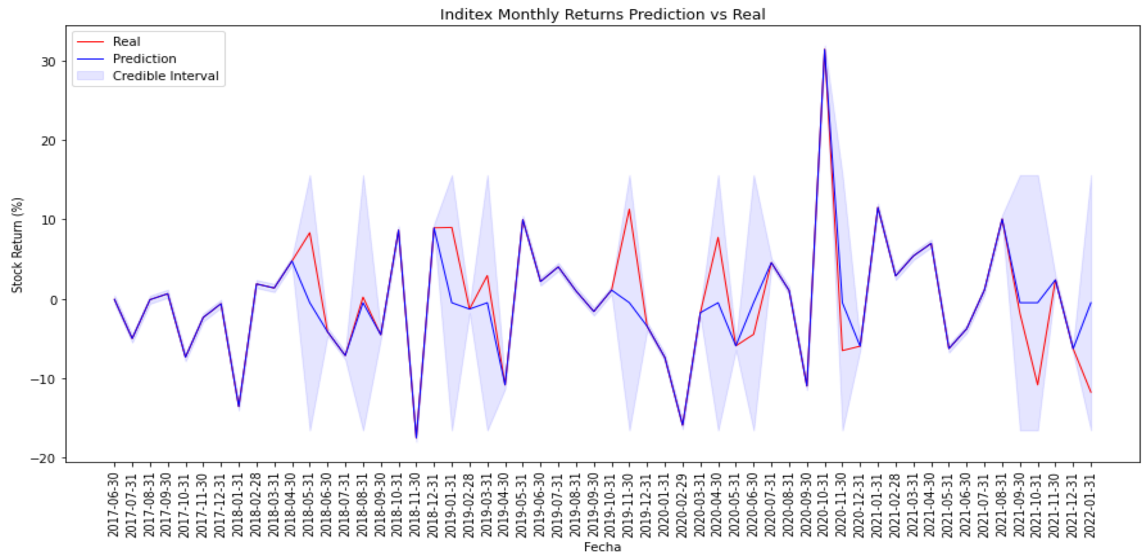

Figure 17: Inditex Monthly Returns Prediction vs Real

[Source: Own elaboration]

As we can see, the model does a very good job of predicting certain values, however, other values show a great variability within the interval of confidence. Apart from that, the model performs very well, only showing significant errors on small occasions.

3.4. Consumer Services

Following the analysis and forecast of the Consumer Goods industry, we will now analyze the Consumer Services industry. More specifically, we will delve into the forecast of the national airport operator, AENA. In Figure 18, we can see AENA´s stock price between January 2017 to June 2022:

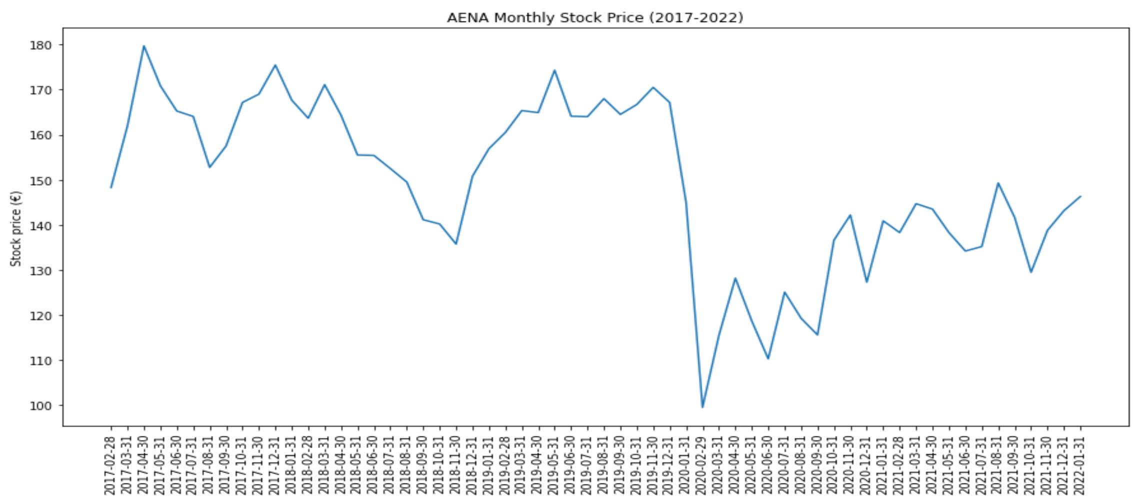

Figure 18: Aena Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

As we can see, AENA´s stock was immensely and drastically impacted by the COVID-19 pandemic, dropping from ~165€ to 100€ in March 2020. As such, it will be important to analyze how the model forecasts such a drop. Following such a price drop, the stock prince suffered up and downs, but, overall, increased steadily. Figure 19 shows an image of the monthly stock returns from the pertinent time interval:

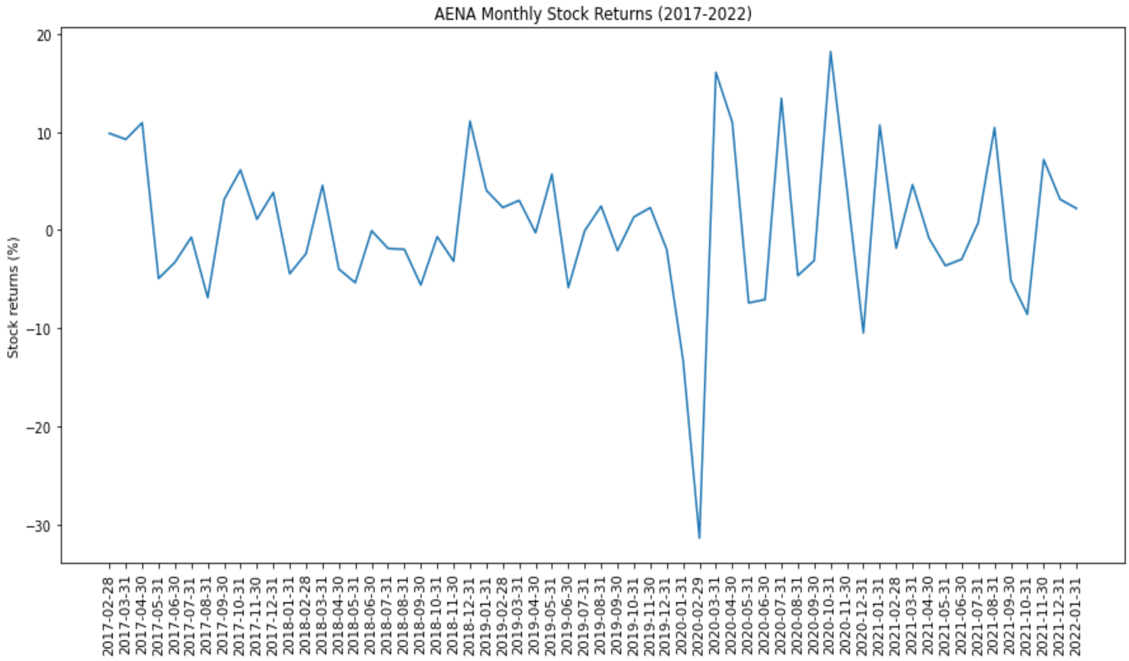

Figure 19: AENA Monthly Stock Returns

[Source: Own elaboration]

As we can see from Figure 19, the returns from 2017 to 2022 were mostly positive, until March 2020, when the COVID-19 pandemic erupted. After that, we can see that returns were very volatile with many ups and downs, however, most ups were more significant than the downs.

In terms of the model, we have opted for a kernel composed of a squared exponential kernel with length scale bounds between 1e-6 and 1e6, and a white noise kernel with noise level bounds between 1e-6 and 1e6. The fitting process of the model to our training results in an RMSE of the testing data of 6,01 and a coefficient of determination, R^2 of 0.966. In this case, contrary to the Consumer Goods industry, we can see that the RMSE is overall satisfactory, while the coefficient of determination is slightly lower than for other industries.

After training and fitting the model, we now use the trained model to predict the entire dataset, to get the overall predictive capabilities of the model. Figure 20 shows the prediction and confidence interval for each month compared to the real values:

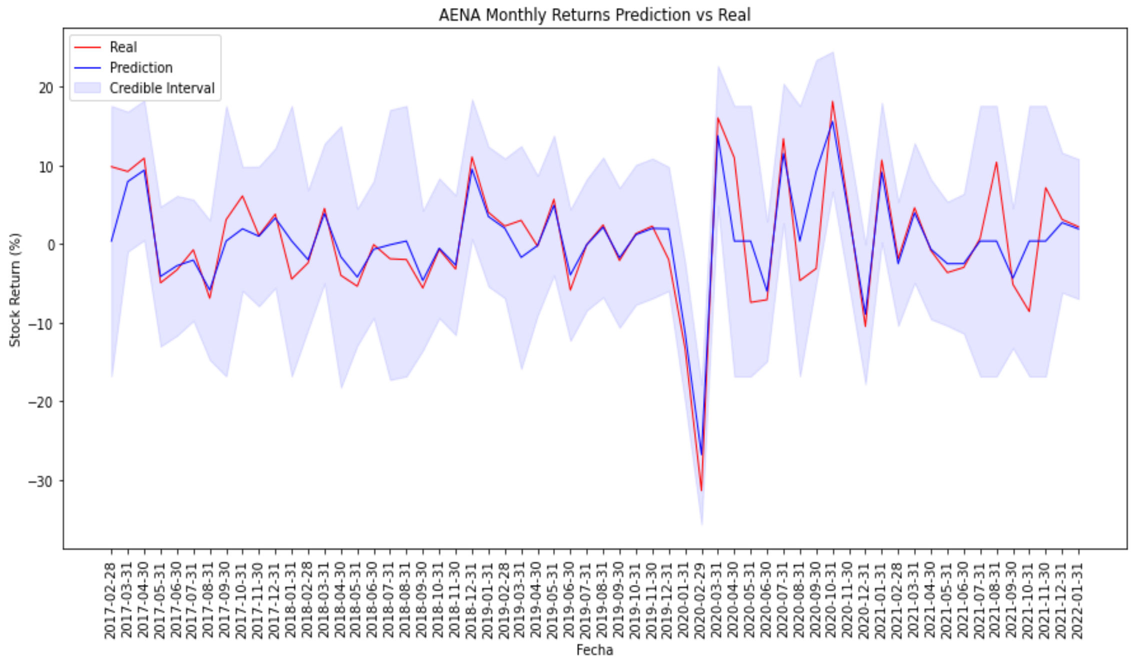

Figure 20: AENA Monthly Returns Prediction vs Real

[Source: Own elaboration]

As we expected from the obtained coefficient of determination, which was lower than for other industries, the prediction is slightly weaker than in previous industries. We can see that certain periods, for example, the years 2021 and 2022, show a higher prediction error, but, overall, the prediction fits the real-time series well. Additionally, we can see that the model did very well in predicting the COVID-19 crash, as well as the previous months leading to such crash.

3.5. Financial Services

The financial services industry is one of the most essential industries for the proper and optimal functioning of the economy. As such, its importance requires comprehensive and adequate analysis. In this case, we have chosen to analyze BBVA, one of the most important banks in Spain, with an extensive international presence. Figure 21 shows BBVA´s monthly stock price between January 2017 and February 2022:

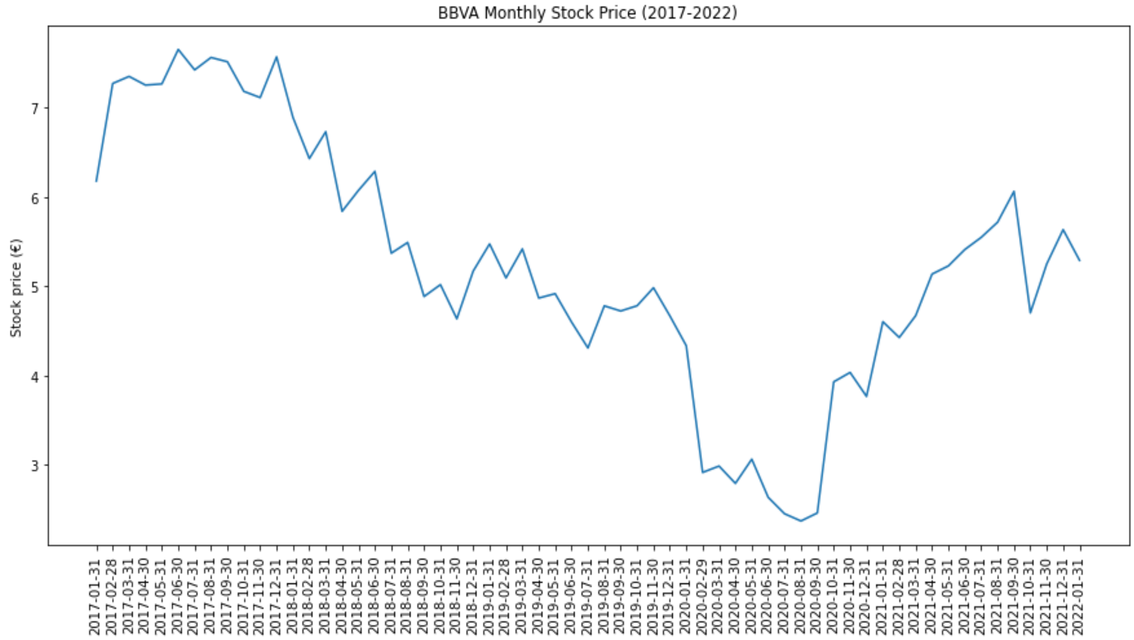

Figure 21: BBVA Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

As we can see, BBVA´s monthly stock price shows a clear downward trend, especially until the year 2021, which showed much better results. These results are not surprising, given the poor performance the financial services industry has shown following the 2008 financial crisis. Figure 22 shows BBVA´s monthly stock returns for the time interval we are analyzing:

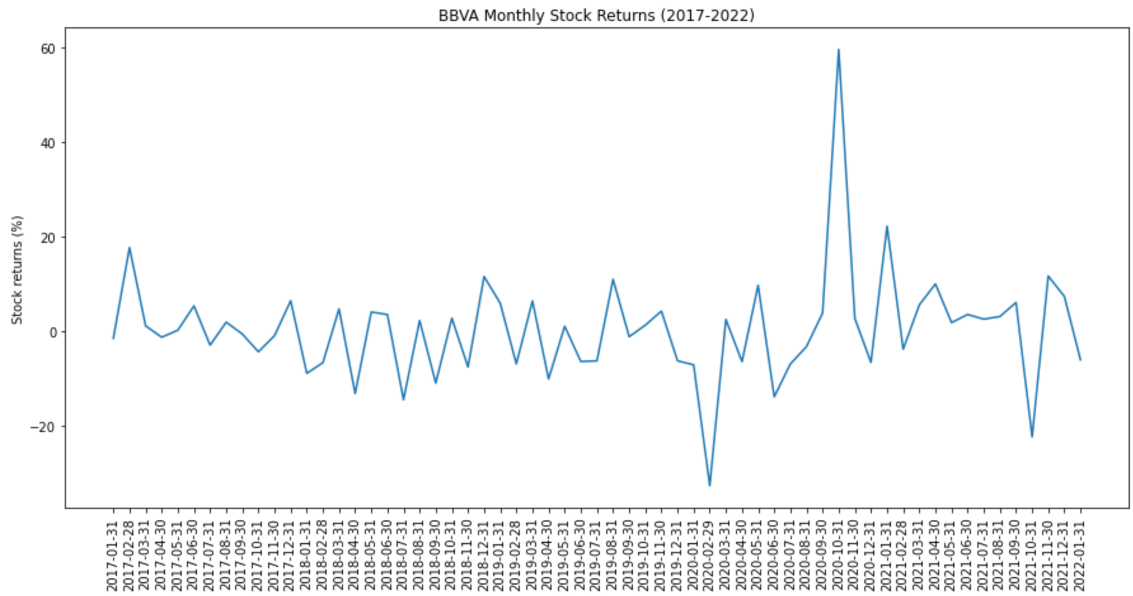

Figure 22: BBVA Monthly Stock Returns

[Source: Own elaboration]

As we can see, the monthly returns from 2017 to 2020 we mostly negative and picked up to show great returns for the years 2021 and 2022.

As with the previous industries, the model will be trained with a training dataset composed of 75% of the available data and will then be evaluated with the testing dataset formed by the remaining 25% of the observations. In terms of the model itself, we have opted for a kernel composed of a squared exponential kernel with length scale bounds between 1e-7 and 1e6 and a white noise kernel with noise level bounds between 1e-6 and 1e6. After fitting the model, we obtain an RMSE of 4,96 and a coefficient of determination of 0,991, both very strong results.

After training the model, we use it to predict the entire dataset and get a summary of the predictive capabilities of the model, and its overall strength. Figure 23 shows the graphical representation of such prediction:

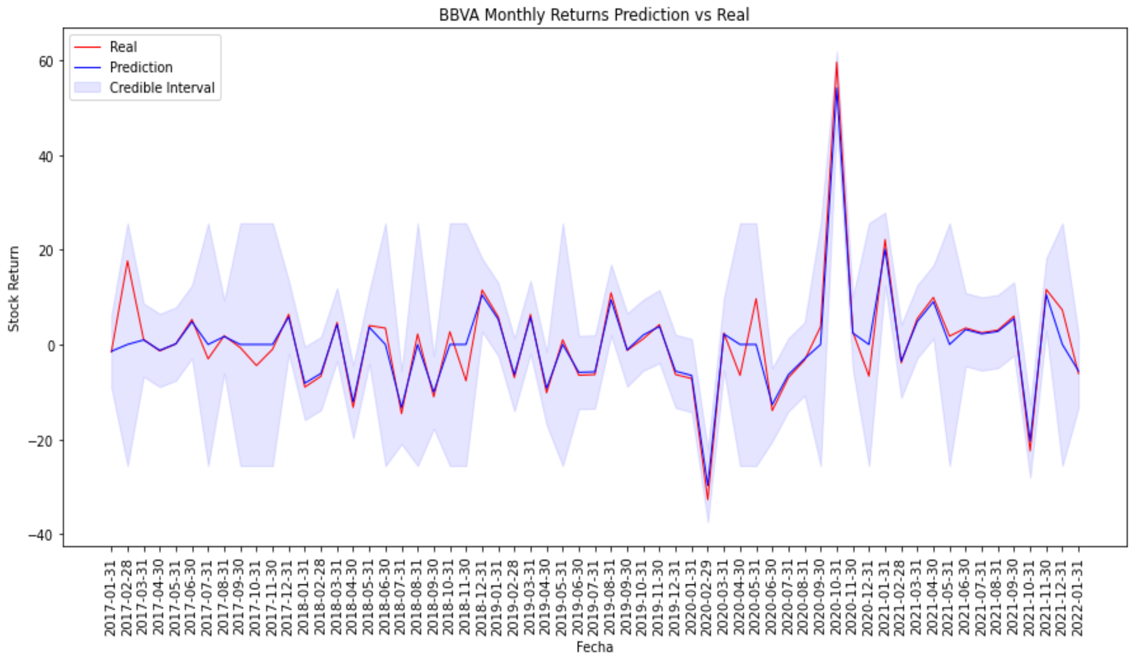

Figure 23: BBVA Monthly Stock Return Prediction vs Real

[Source: Own elaboration]

As we can see from Figure 23, the prediction is very strong on most occasions, and only on certain values does the model differ greatly from the real values. In such cases, the model tends to predict much higher values than the real value. Overall, however, we can assert that the model´s performance is very strong and showed great results.

3.6. Technology and Telecommunications

As with the financial services industry, the technology and telco industry provide essential services to the economy, and, as such, its forecast will be very interesting to analyze. In this case, we will analyze Telefonica, the leading telecommunication company in Spain, and one of the most important companies in Spain. In Figure 24, we can see Telefonica´s monthly stock price between January 2017 and February 2022:

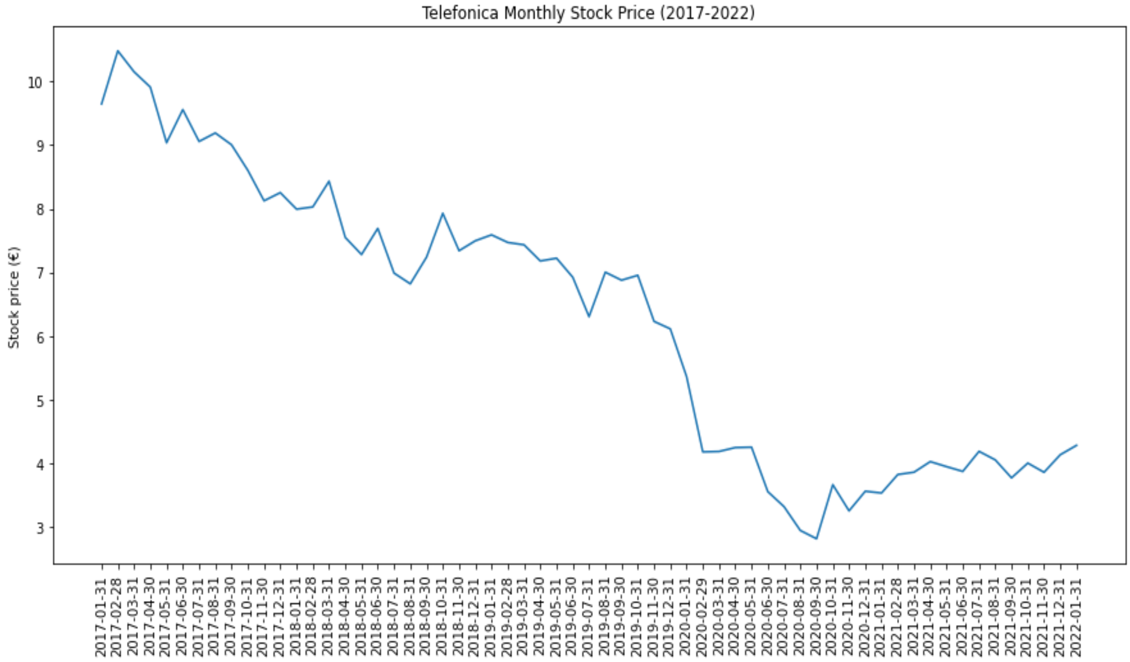

Figure 24: Telefonica Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

As we can see, Telefonica´s stock performance has been very poor, showing a constant downward trend and only showing certain signs of recovery in the year 2021. This, as with the financial services industry, is a constant within the telecommunications industry, and, as such, the model should forecast that accordingly. As with the monthly stock price, we can see the monthly returns in Figure 25:

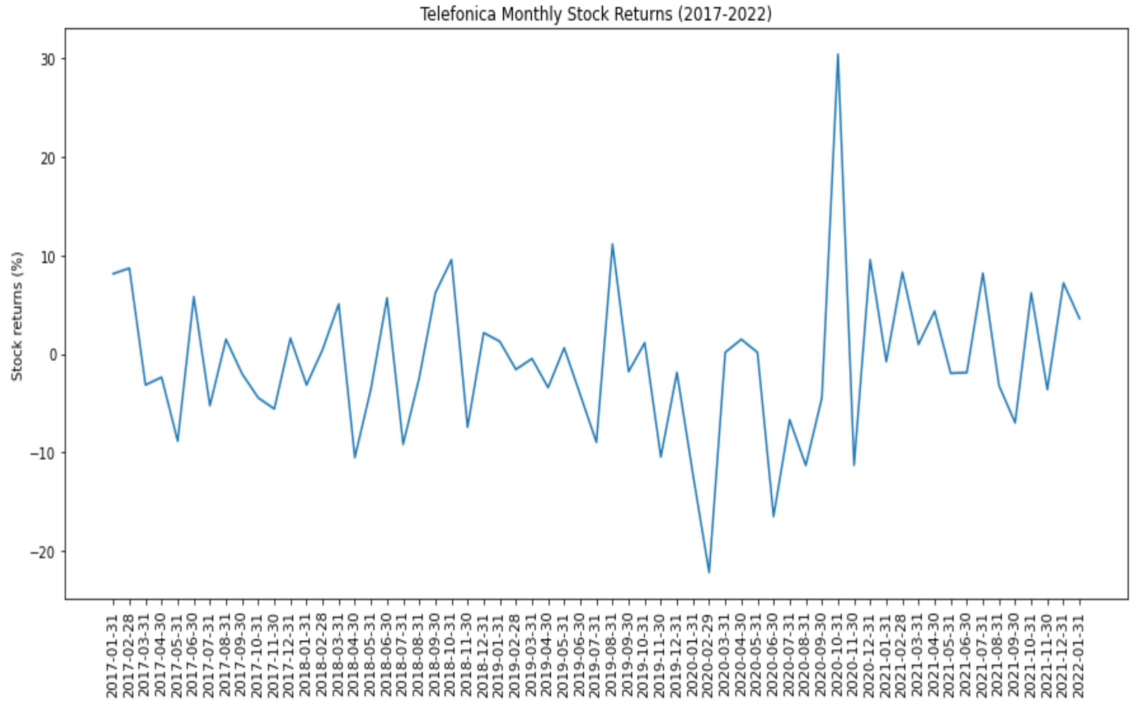

Figure 25: Telefonica Monthly Stock Returns

[Source: Own elaboration]

As expected, the returns are very volatile, and, in general, show a downward trend, which leads to a downward trend in the stock price. In terms of the data partitioning, we have randomly divided the data into a 75% training dataset and a 25% testing dataset.

As with the previous cases, the training dataset will be employed to train the model and the testing dataset will be used to evaluate its performance. After training the model, we obtain an RMSE of 6,07 and a coefficient of determination of 0,992. The obtained results are strong, especially the coefficient of determination, and the slightly higher RMSE might mean that the model struggles with outliers or higher values than expected.

Figure 26 shows the predictions and confidence interval of the entire dataset compared to the real values:

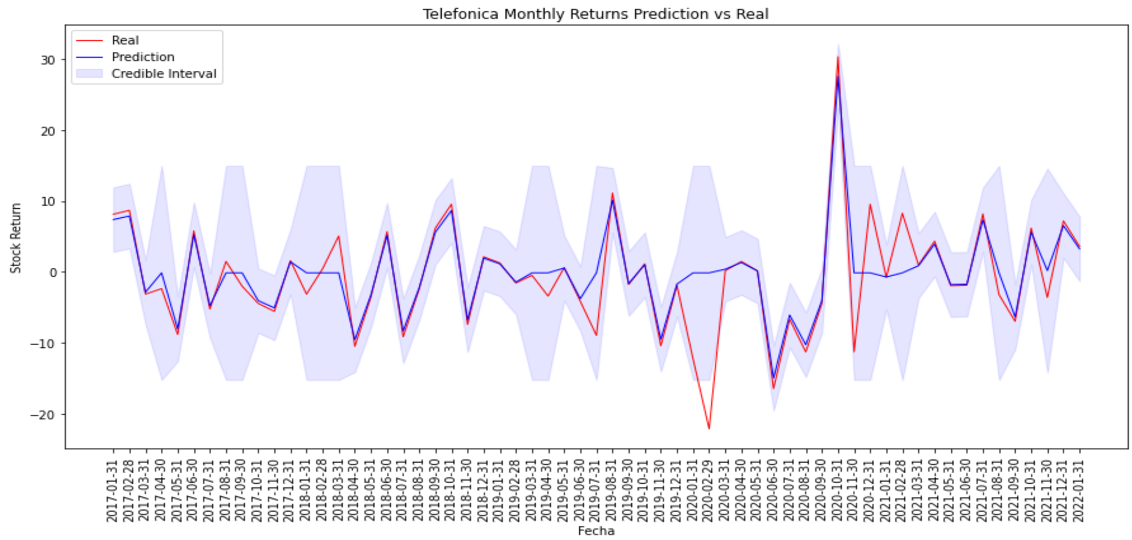

Figure 26: Telefonica Monthly Stock Return Prediction vs Real

[Source: Own elaboration]

As we can see from Figure 26, the overall performance of the model´s prediction is accurate and robust. However, we can see that the model failed to predict the COVID-19 pandemic´s effect on the stock returns. As we stated, the slightly higher RMSE could point to a difficulty when dealing with outliers, which can be confirmed by the difficulty in predicting the pandemic´s impact. Nonetheless, the model portrayed a robust prediction, and the only difficulties were dealing with outliers.

3.7. Real Estate Services

The final industry to be analyzed is the real estate services industry, a very cyclical industry that is dependent on the performance of the economy as a whole. In this case, we have chosen to analyze Merlin Properties, one of the leading real estate investment firms in Spain.

In Figure 27, we can see Merlin´s monthly stock price between January 2017 and February 2022:

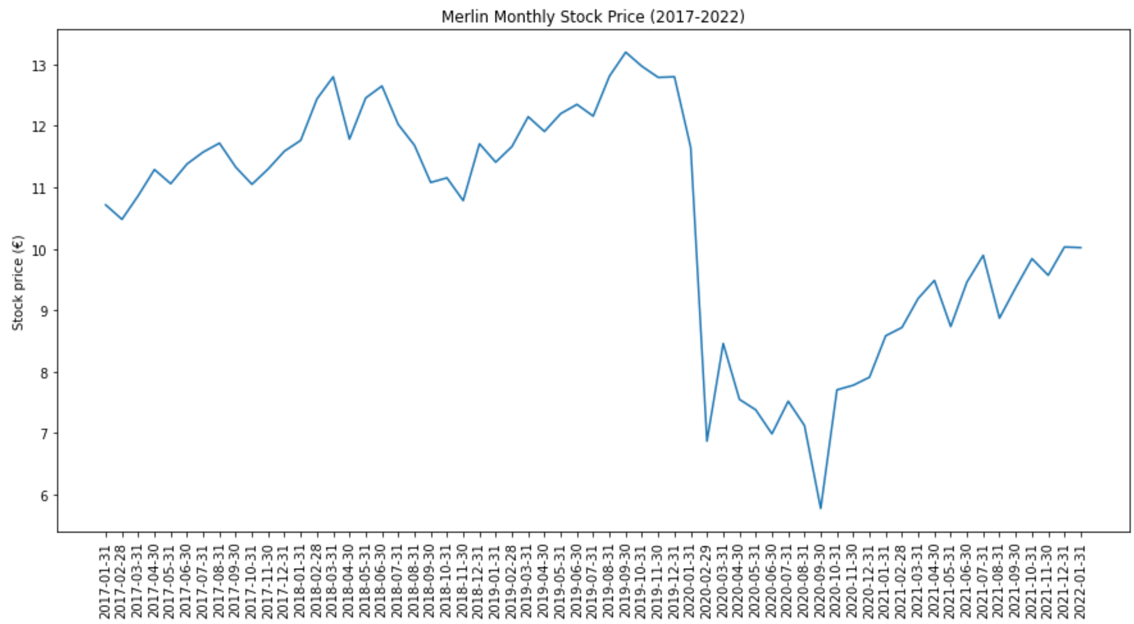

Figure 27: Merlin Monthly Stock Price

[Source: Own elaboration from Yahoo Finance data]

As we can see, the impact of the pandemic was drastic, given that the real estate market depends heavily on the health and strength of the economy. Other than that, Merlin´s stock price shows a clear upwards trend, and even after being setback due to the pandemic, it was able to recover strongly. In Figure 28, we can see the monthly stock returns:

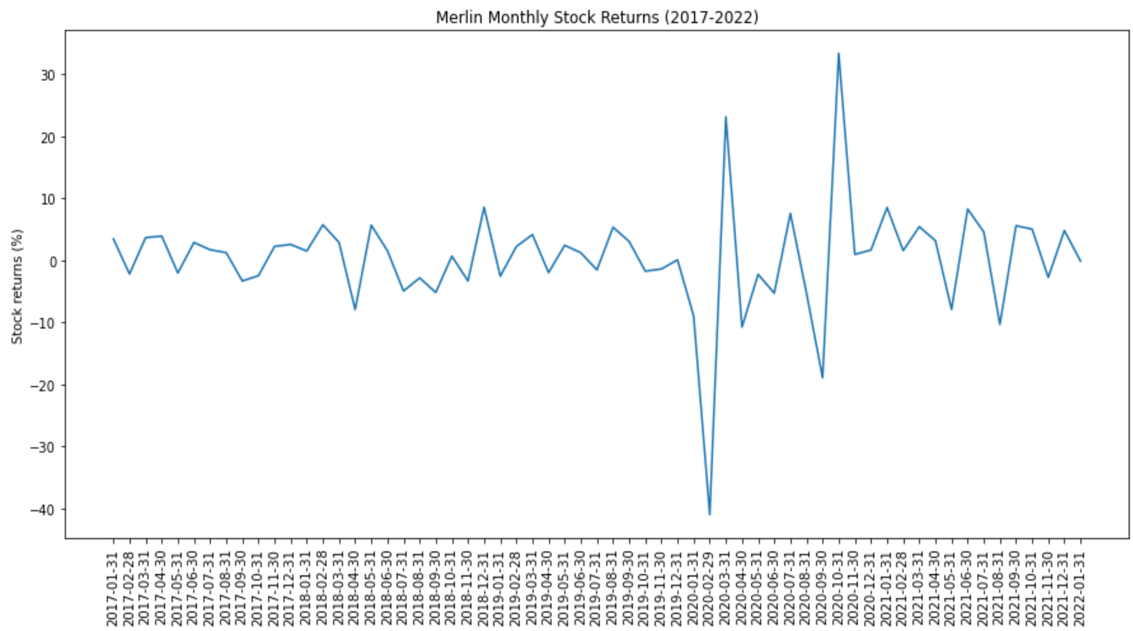

Figure 28: Merlin Monthly Stock Returns

[Source: Own elaboration]

As we can see, the stock returns were very solid, without displaying much volatility, and were mostly constrained within a range. However, the pandemic had a huge impact on the stock market, and, as such, we can see that after the pandemic erupted the returns were much more volatile.

After analyzing the price and returns, we proceed to develop and fit the model. As with the previous industries, we have opted for a training dataset size of 45 monthly observations, equivalent to 75% of the data, and a testing dataset size of 16 monthly observations, equivalent to the remaining 25% of the data. In terms of the chosen kernel, we have opted for a kernel composed of a squared exponential kernel with length scale bounds between 1e-7 and 1e6 and a white noise kernel with noise level bounds between 1e-8 and 1e6. Furthermore, after training the model we obtain an RMSE of 5,84 and a coefficient of determination of 58,22, both of which show a very strong performance. Finally, we use the trained model to predict the entire dataset, which is shown in Figure 29:

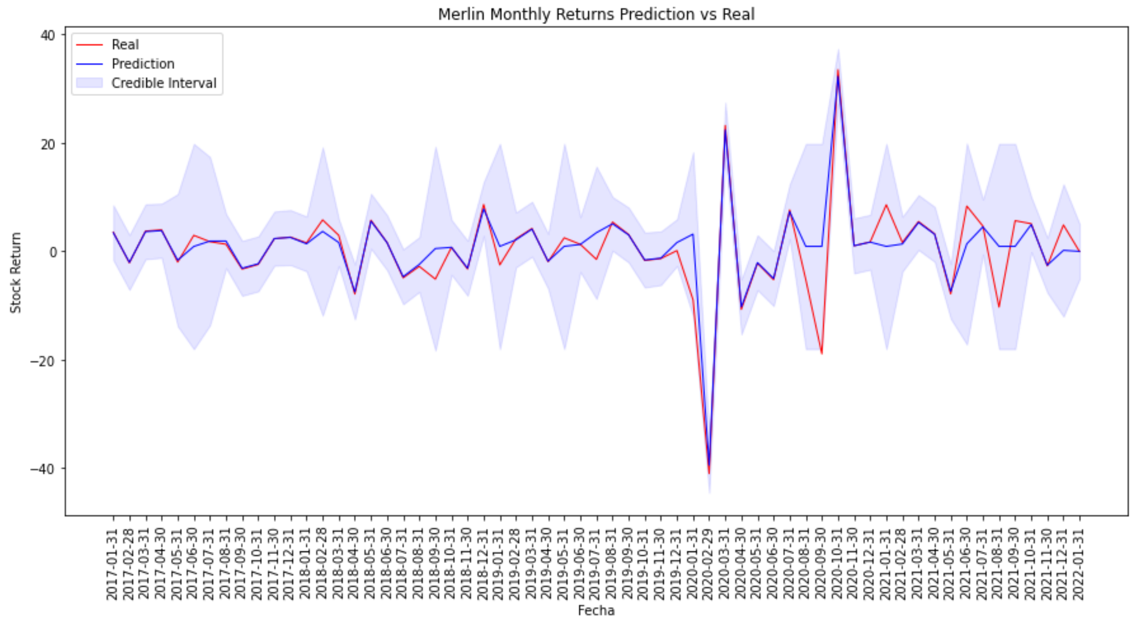

Figure 29: Merlin Monthly Returns Prediction vs Real

[Source: Own elaboration]

As we can see, the prediction´s performance is varied. While initially the model was able to predict the observations with great accuracy, as volatility within the returns grew, the prediction grew weaker, particularly in the years 2020 and 2021. Overall, however, the model shows satisfactory accuracy and performance, and solid robustness.

4. Results Evaluation

To properly assess any model or data evaluation technique, a valid benchmark must be used, particularly when the task at hand involves forecasting or prediction, as is the case when dealing with Gaussian process models. In this case, the chosen benchmark will be a Support Vector Machine regression model, one of the most typical Machine Learning models available.

Support Vector Machines are one of the most common and widely used supervised learning models. Support vector machines were first developed as a classification method that aims to learn by finding a hyperplane that clearly classifies a set of observations or data points. However, they were later modified to be able to handle regression problems, and, since then, they have become one of the most widely used regression models. Furthermore, they provide certain advantages, such as handling high dimensionality problems or their versatility.

To provide an accurate and valid comparison to Gaussian Process models, Support Vector Machine regression models have been fit to the same data employed in the training and testing of Gaussian Process models.

Table 1 shows a summary of the results obtained by the Gaussian process models in comparison to Support Vector Machine regression models:

	Gaussian Process		*SVM Regression*	
	RMSE	**Coefficient of Determination**	**RMSE**	**Coefficient of Determination**
Iberdrola	3,24	0,992	4,98	0,718
ACS	5,46	0,992	7,89	0,492
Inditex	6,85	0,999	7,01	0,480
AENA	6,01	0,966	5,58	0,708
BBVA	4,96	0,991	5,97	0,703
Telefonica	6,07	0,992	5,96	0,820
Merlin	5,84	0,997	3,94	0,549

Table 1: Results Comparison GP Model vs SVM

[Source: Own elaboration]

As we can see from Table 1, the results obtained by the Gaussian process models are exceptional, especially when comparing them to those obtained by the Support Vector Machine regression model. We can extract several conclusions from the table above. Firstly, we can see that in a lot of cases, such as Iberdrola, ACS, Inditex, or BBVA, the root mean squared error is lower when using Gaussian Process models. This means that, on average, the standard deviation of the prediction errors is lower than the used benchmark, the Support Vector Machine regression model. Furthermore, even in the cases in which the root mean squared error is lower for the Support Vector Machine regressor the difference is minimal, except in the case of Merlin which shows a bigger disparity.

Nonetheless, the most glaring and crucial conclusion we can extract is derived from the coefficient of determination. The difference between the two models is immense, and we can see that despite a similarity regarding the root mean squared error, Gaussian process models perform much better in learning the available data, and, ultimately the model fit is much more precise and accurate. As such, the predictive capacity of Gaussian models is much superior to that of Support Vector Machine regression models, and, in the future, and long run, would lead to much better results.

All in all, when compared to the employed benchmark, Support Vector Machine regression models, Gaussian Process models have demonstrated remarkable performance, showing great predictive capabilities, and unique versatility in their adaptability to different types of time series, and, in this case, companies from different sectors.

5. Conclusions

T he developed project has allowed for a deep, exhaustive exploration of Gaussian process models, and their application to financial time series. Derived from the project we have ascertained that Gaussian process models have performed successfully throughout the different sectors of the Spanish stock market, and they have, therefore, demonstrated their versatility.

In terms of the structure of the dissertation, it was divided into three phases. The first of those phases revolved around Gaussian process models, and Bayesian modeling in general. Gaussian process models are a very complex and intricate type of model, and, as such, require a full understanding of their mathematical and conceptual foundation. Moreover, given their non-parametric nature, a basic understanding of the concepts behind Bayesian modeling is of great pertinence and can lead to a much smoother and more successful application of the models. As well as understanding the conceptual basis of Gaussian process models, diving into the different components involved in the modeling process is of the utmost importance and its comprehension is imperative to ensure a successful application of Gaussian process models.

The second phase of the dissertation involved the analysis of the Spanish business and economic community, and the different sectors involved in it. To extract valid and meaningful conclusions regarding the application of Gaussian process models to forecast financial time series, the proper scope of the analysis must be determined. As such, an extensive analysis must be performed to narrow down the wide variety of companies on the stock market. As part of that analysis, the sectorial classification employed by the Spanish Stock Exchange (BME) was used, and, from such classification, a list of representative companies was derived.

With a list of companies narrowed down and the conceptual foundation of Gaussian process models understood, the third phase of the dissertation was the practical application of the models to the chosen companies. The application and development of the Gaussian process models were performed using Python, and more specifically Scikit Learn (SkLearn), one of its public Machine Learning libraries. In such an application process, data was first extracted using the Yahoo Financials library and then divided into a training and testing dataset. After that, a model based on Gaussian processes is trained using the training dataset and evaluated using the testing dataset.

Throughout that process, some conclusions were extracted. Firstly, several different kernels were tried, and the differences in terms of RMSE were mostly minimal. However, the differences were more noticeable in the coefficient of determination, a measure that indicates how well the model replicates observations. Additionally, the results obtained demonstrated great versatility and adaptability to different time series of financial nature. Table 2 shows a summary of the obtained results and measures by the Gaussian Process model for each company/sector:

	RMSE	**Coefficient of Determination**
Iberdrola	3,24	0,992
ACS	5,46	0,992
Inditex	6,85	0,999
AENA	6,01	0,966
BBVA	4,96	0,991
Telefonica	6,07	0,992
Merlin	5,84	0,997

Table 2: Results Summary Gaussian Process Model

[Source: Own elaboration]

As we can see from Table 2, the obtained results demonstrate that Gaussian process models not only demonstrate great forecasting accuracy but also a great adaptative capacity, displaying exceptional results with stocks from a wide variety of sectors. As such, we were able to ascertain the predictive capacity of Gaussian process models, and, through this thesis, an initial approach to their application was performed. Furthermore, Support Vector Machine regression models were used as a benchmark, and, once again, Gaussian Process models showed their value. They demonstrated superior versatility in their capacity to predict different time series, and, overall, the Gaussian process models fit much better with the available data. As such, it is clear that Gaussian process models have significant value and should be further used and applied in forecasting tasks.

Finally, while this project was an extensive first approach to the application of Gaussian process models to the forecasting of financial time series, certain areas of study could be of interest to further analyze its strengths, weaknesses, and possible areas of use. Amongst such future areas of research is the use of Deep Gaussian Process models, an extension of the classic Gaussian Process regression model. Deep Gaussian Process models employ deep learning techniques in the construction of kernels, and, as such, can capture complex data while maintaining probabilistic inference. The application of Deep Gaussian Process models to financial markets was first approached by Shi, Dai, Long & Li (2021), and they concluded that a combination of a Gaussian Process model with an LSTM kernel showed superior performance compared to the chosen benchmarks. As such, a deep analysis of the predictive capabilities of Deep Gaussian Process models would greatly complement the conclusions derived from this dissertation.

Bibliography

BME (2022). *Bolsa de Madrid - Listado de Empresas por Sectores*. (2022). https://www.bolsamadrid.es/esp/aspx/Empresas/EmpresasPorSectores.aspx

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions*, *7*(1), 1525-1534.

Corporate Finance Institute. (2022, February 4). *Stock Analysis*. https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/stock-analysis/

Duvenaud, D. K. (2014, June). *Automatic Model Construction with Gaussian Processes*. https://doi.org/10.17863/CAM.14087

Gandhmal, D. P., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, *34*, 100190. https://doi.org/10.1016/j.cosrev.2019.08.001

Gurudath, S. (2022, March 30). *Parametric and Non-parametric Models In Machine Learning*. Medium. https://medium.com/analytics-vidhya/parametric-and-nonparametric-models-in-machine-learning-a9f63999e233

Liu, H., Ong, Y. S., Shen, X., & Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE transactions on neural networks and learning systems*, *31*(11), 4405-4423.

Natsume, Y. (2022, May 7). *Gaussian Process Kernels - Towards Data Science*. Medium. https://towardsdatascience.com/gaussian-process-kernels-96bafb4dd63e

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), 3007-3057.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine learning* (pp. 101-121). Academic Press.

Rasmussen, C. E., & Williams, C. K., I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Renaud, O., & Victoria-Feser, M. P. (2010). A robust coefficient of determination for regression. *Journal of Statistical Planning and Inference*, *140*(7), 1852-1862.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., & Aigrain, S. (2013). Gaussian processes for time series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*(1984), 20110550.

Sánchez, Á. (2021, January 27). *Una legión de foreros de Reddit hace perder miles de millones a fondos bajistas de Wall Street*. El País. https://elpais.com/economia/2021-01-27/una-legion-de-foreros-de-reddit-hace-perder-miles-de-millones-a-fondos-bajistas-de-wall-street.html

Schulz, E., Speekenbrink, M., & Krause, A. (2018, August). *A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions*. https://doi.org/10.1016/j.jmp.2018.03.001

SIFMA. (2021). *Capital Markets Fact Book.* https://www.sifma.org/resources/research/fact-book/

Scikit-Learn. *sklearn.gaussian_process.GaussianProcessRegressor*. (2022). https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html

Scikit-Learn. sklearn.svm.SVR. (2022). https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR

Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, *7*(2), 26.

Shi, Y., Dai, W., Long, W., & Li, B. (2021). Deep Kernel Gaussian Process Based Financial Market Predictions. *arXiv preprint arXiv:2105.12293*.

Suthaharan, S. (2016). *Support Vector Machine. In: Machine Learning Models and Algorithms for Big Data Classification*. Integrated Series in Information Systems, vol 36. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7641-3_9

van Boom, D. (2022, May 25). *Luna Crypto Crash: How UST Broke and What's Next for Terra*. CNET. https://www.cnet.com/personal-finance/crypto/luna-crypto-crash-how-ust-broke-and-whats-next-for-terra/

Wilson, A. G., Knowles, D. A., & Ghahramani, Z. (2011). Gaussian process regression networks. *arXiv preprint arXiv:1110.4411*.

Appendix

In the following appendix lies the detailed code for each model:

Energy and Oil Model

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

# # Extracción Datos

yahoo_financials = YahooFinancials('IBE.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
ib_monthly = pd.DataFrame(data['IBE.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
ib_monthly.head()

fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(ib_monthly.index, ib_monthly['close'])
pl.title('Iberdrola Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

ib_monthly['Rentabilidad'] = ib_monthly['close'].pct_change(1)
ib_monthly['Rentabilidad'] = ib_monthly['Rentabilidad'].apply(lambda
x: x*100)
ib_monthly = ib_monthly.dropna()
ib_monthly

fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(ib_monthly.index, ib_monthly['Rentabilidad'])
pl.title('Iberdrola Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock Return (%)')

train_n = random.sample(range(0, len(ib_monthly['Rentabilidad'])), 45)
ib_monthly_tr = ib_monthly.iloc[train_n]
ib_monthly_test = ib_monthly.drop(ib_monthly.index[train_n])
```

```
print(ib_monthly_tr.shape, ib_monthly_test.shape)

ib_monthly_x_tr = ib_monthly_tr['close'].to_numpy().reshape(-1, 1)
ib_monthly_y_tr = ib_monthly_tr['Rentabilidad'].to_numpy().reshape(-1,
1)
ib_monthly_x_test = ib_monthly_test['close'].to_numpy().reshape(-1, 1)
ib_monthly_y_test =
ib_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression

kernel = gp.kernels.WhiteKernel(1, (1e-5, 1e5)) + gp.kernels.RBF(1,
(1e-5, 1e5))
model_ib = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=10, alpha=0.1, normalize_y=True)
model_ib.fit(ib_monthly_x_tr, ib_monthly_y_tr)
params_ib = model_ib.kernel_.get_params()

model_ib.score(ib_monthly_x_tr, ib_monthly_y_tr)

y_pred_ib, std_ib = model_ib.predict(ib_monthly_x_test,
return_std=True)
MSE_ib = ((y_pred_ib-ib_monthly_y_test)**2).mean()
MSE_ib


rmse = np.sqrt((y_pred_ib-ib_monthly_y_test)**2).mean()
rmse

ib_monthly_x_tot = ib_monthly['close'].to_numpy().reshape(-1, 1)
ib_monthly_y_tot = ib_monthly['Rentabilidad'].to_numpy().reshape(-1,
1)

y_pred_ib_total, std_ib_total = model_ib.predict(ib_monthly_x_tot,
return_std=True)
ib_monthly['PrediccíonTotal'] = ib_monthly['Rentabilidad']
ib_monthly.iloc[:, 2] = y_pred_ib_total[:,0]
ib_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(ib_monthly.index, ib_monthly['Rentabilidad'], 'r-', linewidth
= 1, label=u'Real')
pl.plot(ib_monthly.index, ib_monthly['PrediccíonTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=ib_monthly.index,
    y1=(ib_monthly['PrediccíonTotal']-1.96*std_ib_total),
    y2=(ib_monthly['PrediccíonTotal']+1.96*std_ib_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Date')
pl.xticks(rotation=90)
pl.ylabel('Stock Return')
pl.legend(loc="upper left")
pl.title('Iberdrola Monthly Returns Prediction vs Real')
```

Construction Model

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

# # Extracción Datos

yahoo_financials = YahooFinancials('ACS.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
acs_monthly = pd.DataFrame(data['ACS.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
acs_monthly.head()

fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(acs_monthly.index, acs_monthly['close'])
pl.title('ACS Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

acs_monthly['Rentabilidad'] = acs_monthly['close'].pct_change(1)
acs_monthly['Rentabilidad'] = acs_monthly['Rentabilidad'].apply(lambda
x: x*100)
acs_monthly = acs_monthly.dropna()
acs_monthly

fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(acs_monthly.index, acs_monthly['Rentabilidad'])
pl.title('ACS Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock Return (%)')

train_n = random.sample(range(0, len(acs_monthly['Rentabilidad'])),
45)
acs_monthly_tr = acs_monthly.iloc[train_n]
acs_monthly_test = acs_monthly.drop(acs_monthly.index[train_n])
print(acs_monthly_tr.shape, acs_monthly_test.shape)

acs_monthly_x_tr = acs_monthly_tr['close'].to_numpy().reshape(-1, 1)
```

```
acs_monthly_y_tr = acs_monthly_tr['Rentabilidad'].to_numpy().reshape(-
1, 1)
acs_monthly_x_test = acs_monthly_test['close'].to_numpy().reshape(-1,
1)
acs_monthly_y_test =
acs_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression

kernel = gp.kernels.WhiteKernel(1, (1e-6, 1e6)) + gp.kernels.RBF(1,
(1e-6, 1e6))
model_acs = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=10, alpha=0.1, normalize_y=True)
model_acs.fit(acs_monthly_x_tr, acs_monthly_y_tr)
params_acs = model_acs.kernel_.get_params()

model_acs.score(acs_monthly_x_tr, acs_monthly_y_tr)

y_pred_acs, std_acs = model_acs.predict(acs_monthly_x_test,
return_std=True)
MSE_acs = ((y_pred_acs-acs_monthly_y_test)**2).mean()
MSE_acs

rmse = np.sqrt((y_pred_acs-acs_monthly_y_test)**2).mean()
rmse

acs_monthly_x_tot = acs_monthly['close'].to_numpy().reshape(-1, 1)
acs_monthly_y_tot = acs_monthly['Rentabilidad'].to_numpy().reshape(-1,
1)

y_pred_acs_total, std_acs_total = model_acs.predict(acs_monthly_x_tot,
return_std=True)
acs_monthly['PrediccíónTotal'] = acs_monthly['Rentabilidad']
acs_monthly.iloc[:, 2] = y_pred_acs_total[:,0]
acs_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(acs_monthly.index, acs_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(acs_monthly.index, acs_monthly['PrediccíónTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=acs_monthly.index,
    y1=(acs_monthly['PrediccíónTotal']-1.96*std_acs_total),
    y2=(acs_monthly['PrediccíónTotal']+1.96*std_acs_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Stock Return (%)')
pl.legend(loc="upper left")
pl.title('ACS Monthly Returns Prediction vs Real')
```

Consumer Goods Model

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

yahoo_financials = YahooFinancials('ITX.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-06-
01',
                                                   end_date='2022-02-
15',

time_interval='monthly')
ind_monthly = pd.DataFrame(data['ITX.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
ind_monthly.head()



fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(ind_monthly.index, ind_monthly['close'])
pl.title('Inditex Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')


ind_monthly['Rentabilidad'] = ind_monthly['close'].pct_change(1)
ind_monthly['Rentabilidad'] = ind_monthly['Rentabilidad'].apply(lambda
x: x*100)
ind_monthly = ind_monthly.dropna()
ind_monthly

fig = pl.figure(figsize = (15,5))
pl.xticks(rotation=90)
pl.plot(ind_monthly.index, ind_monthly['Rentabilidad'])
pl.title('Inditex Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock Return (%)')

train_n = random.sample(range(0, len(ind_monthly['Rentabilidad'])),
45)
ind_monthly_tr = ind_monthly.iloc[train_n]
ind_monthly_test = ind_monthly.drop(ind_monthly.index[train_n])

print(ind_monthly_tr.shape, ind_monthly_test.shape)
ind_monthly_x_tr = ind_monthly_tr['close'].to_numpy().reshape(-1, 1)
```

```
ind_monthly_y_tr = ind_monthly_tr['Rentabilidad'].to_numpy().reshape(-
1, 1)
ind_monthly_x_test = ind_monthly_test['close'].to_numpy().reshape(-1,
1)
ind_monthly_y_test =
ind_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression

kernel = gp.kernels.WhiteKernel(1, (1e-8, 1e6)) + gp.kernels.RBF(1,
(1e-7, 1e6))
model_ind = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=20, alpha=0.001, normalize_y=True)
model_ind.fit(ind_monthly_x_tr, ind_monthly_y_tr)
params_ind = model_ind.kernel_.get_params()

model_ind.score(ind_monthly_x_tr, ind_monthly_y_tr)

y_pred_ind, std_ind = model_ind.predict(ind_monthly_x_test,
return_std=True)
MSE_ind = ((y_pred_ind-ind_monthly_y_test)**2).mean()
MSE_ind

rmse = np.sqrt((y_pred_ind-ind_monthly_y_test)**2).mean()
rmse

ind_monthly_x_tot = ind_monthly['close'].to_numpy().reshape(-1, 1)
ind_monthly_y_tot = ind_monthly['Rentabilidad'].to_numpy().reshape(-1,
1)

y_pred_ind_total, std_ind_total = model_ind.predict(ind_monthly_x_tot,
return_std=True)
ind_monthly['PredicciónTotal'] = ind_monthly['Rentabilidad']
ind_monthly.iloc[:, 2] = y_pred_ind_total[:,0]
ind_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(ind_monthly.index, ind_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(ind_monthly.index, ind_monthly['PredicciónTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=ind_monthly.index,
    y1=(ind_monthly['PredicciónTotal']-1.96*std_ind_total),
    y2=(ind_monthly['PredicciónTotal']+1.96*std_ind_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Rentabilidad Mensual')
pl.ylabel('Stock Return (%)')
pl.legend(loc="upper left")
pl.title('Inditex Monthly Returns Prediction vs Real')
```

Consumer Services Model

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat

# # Extracción Datos

# In[2]:
yahoo_financials = YahooFinancials('AENA.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                    end_date='2022-02-
15',

time_interval='monthly')
aena_monthly = pd.DataFrame(data['AENA.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
aena_monthly.head()

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(aena_monthly.index, aena_monthly['close'])
pl.title('AENA Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

aena_monthly['Rentabilidad'] = aena_monthly['close'].pct_change(1)
aena_monthly['Rentabilidad'] =
aena_monthly['Rentabilidad'].apply(lambda x: x*100)
aena_monthly = aena_monthly.dropna()
aena_monthly

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(aena_monthly.index, aena_monthly['Rentabilidad'])
pl.title('AENA Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock returns (%)')

import random
train_n = random.sample(range(0, len(aena_monthly['Rentabilidad'])),
45)
aena_monthly_tr = aena_monthly.iloc[train_n]
aena_monthly_test = aena_monthly.drop(aena_monthly.index[train_n])

print(aena_monthly_tr.shape, aena_monthly_test.shape)

aena_monthly_x_tr = aena_monthly_tr['close'].to_numpy().reshape(-1, 1)
aena_monthly_y_tr =
aena_monthly_tr['Rentabilidad'].to_numpy().reshape(-1, 1)
```

```
aena_monthly_x_test = aena_monthly_test['close'].to_numpy().reshape(-
1, 1)
aena_monthly_y_test =
aena_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression

kernel = gp.kernels.WhiteKernel(1, (1e-6, 1e6)) + gp.kernels.RBF(1,
(1e-6, 1e6))
model_aena = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=20, alpha=0.001, normalize_y=True)
model_aena.fit(aena_monthly_x_tr, aena_monthly_y_tr)
params_aena = model_aena.kernel_.get_params()

model_aena.score(aena_monthly_x_tr, aena_monthly_y_tr)

y_pred_aena, std_aena = model_aena.predict(aena_monthly_x_test,
return_std=True)
MSE_aena = ((y_pred_aena-aena_monthly_y_test)**2).mean()
MSE_aena

rmse = np.sqrt((y_pred_aena-aena_monthly_y_test)**2).mean()
rmse

aena_monthly_x_tot = aena_monthly['close'].to_numpy().reshape(-1, 1)
aena_monthly_y_tot = aena_monthly['Rentabilidad'].to_numpy().reshape(-
1, 1)

y_pred_aena_total, std_aena_total =
model_aena.predict(aena_monthly_x_tot, return_std=True)
aena_monthly['PrediccionTotal'] = aena_monthly['Rentabilidad']
aena_monthly.iloc[:, 2] = y_pred_aena_total[:,0]
aena_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(aena_monthly.index, aena_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(aena_monthly.index, aena_monthly['PrediccionTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=aena_monthly.index,
    y1=(aena_monthly['PrediccionTotal']-1.96*std_aena_total),
    y2=(aena_monthly['PrediccionTotal']+1.96*std_aena_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Rentabilidad Mensual')
pl.ylabel('Stock Return (%)')
pl.legend(loc="upper left")
pl.title('AENA Monthly Returns Prediction vs Real')
```

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

# # Extracción Datos
yahoo_financials = YahooFinancials('BBVA.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
bbva_monthly = pd.DataFrame(data['BBVA.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
bbva_monthly.head()

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(bbva_monthly.index, bbva_monthly['close'])
pl.title('BBVA Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

bbva_monthly['Rentabilidad'] = bbva_monthly['close'].pct_change(1)
bbva_monthly['Rentabilidad'] =
bbva_monthly['Rentabilidad'].apply(lambda x: x*100)
bbva_monthly = bbva_monthly.dropna()
bbva_monthly

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(bbva_monthly.index, bbva_monthly['Rentabilidad'])
pl.title('BBVA Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock returns (%)')

train_n = random.sample(range(0, len(bbva_monthly['Rentabilidad'])),
45)
bbva_monthly_tr = bbva_monthly.iloc[train_n]
bbva_monthly_test = bbva_monthly.drop(bbva_monthly.index[train_n])
print(bbva_monthly_tr.shape, bbva_monthly_test.shape)

bbva_monthly_x_tr = bbva_monthly_tr['close'].to_numpy().reshape(-1, 1)
bbva_monthly_y_tr =
bbva_monthly_tr['Rentabilidad'].to_numpy().reshape(-1, 1)
bbva_monthly_x_test = bbva_monthly_test['close'].to_numpy().reshape(-
1, 1)
```

```
bbva_monthly_y_test =
bbva_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression
kernel = gp.kernels.WhiteKernel(1, (1e-6, 1e6)) + gp.kernels.RBF(1,
(1e-7, 1e6))
model_m = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=20, alpha=0.1, normalize_y=True)
model_m.fit(bbva_monthly_x_tr, bbva_monthly_y_tr)
params_m = model_m.kernel_.get_params()

model_m.score(bbva_monthly_x_tr, bbva_monthly_y_tr)

y_pred_m, std = model_m.predict(bbva_monthly_x_test, return_std=True)
MSE_m = ((y_pred_m-bbva_monthly_y_test)**2).mean()
MSE_m

rmse = np.sqrt((y_pred_m-bbva_monthly_y_test)**2).mean()
rmse

bbva_monthly_x_tot = bbva_monthly['close'].to_numpy().reshape(-1, 1)
bbva_monthly_y_tot = bbva_monthly['Rentabilidad'].to_numpy().reshape(-
1, 1)

y_pred_bbva_total, std_bbva_total =
model_m.predict(bbva_monthly_x_tot, return_std=True)
bbva_monthly['PrediccionTotal'] = bbva_monthly['Rentabilidad']
bbva_monthly.iloc[:, 2] = y_pred_bbva_total[:,0]
bbva_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(bbva_monthly.index, bbva_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(bbva_monthly.index, bbva_monthly['PrediccionTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=bbva_monthly.index,
    y1=(bbva_monthly['PrediccionTotal']-1.96*std_bbva_total),
    y2=(bbva_monthly['PrediccionTotal']+1.96*std_bbva_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Rentabilidad Mensual')
pl.ylabel('Stock Return')
pl.legend(loc="upper left")
pl.title('BBVA Monthly Returns Prediction vs Real')
```

Technology and Telecommunications

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
```

```
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

# # Extracción Datos

yahoo_financials = YahooFinancials('TEF.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                   end_date='2022-02-
15',

time_interval='monthly')
tlf_monthly = pd.DataFrame(data['TEF.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
tlf_monthly.head()

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(tlf_monthly.index, tlf_monthly['close'])
pl.title('Telefonica Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

tlf_monthly['Rentabilidad'] = tlf_monthly['close'].pct_change(1)
tlf_monthly['Rentabilidad'] = tlf_monthly['Rentabilidad'].apply(lambda
x: x*100)
tlf_monthly = tlf_monthly.dropna()
tlf_monthly

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(tlf_monthly.index, tlf_monthly['Rentabilidad'])
pl.title('Telefonica Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock returns (%)')


train_n = random.sample(range(0, len(tlf_monthly['Rentabilidad'])),
45)
tlf_monthly_tr = tlf_monthly.iloc[train_n]
tlf_monthly_test = tlf_monthly.drop(tlf_monthly.index[train_n])

print(tlf_monthly_tr.shape, tlf_monthly_test.shape)

tlf_monthly_x_tr = tlf_monthly_tr['close'].to_numpy().reshape(-1, 1)
tlf_monthly_y_tr = tlf_monthly_tr['Rentabilidad'].to_numpy().reshape(-
1, 1)
tlf_monthly_x_test = tlf_monthly_test['close'].to_numpy().reshape(-1,
1)
tlf_monthly_y_test =
tlf_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression
```

```
kernel = gp.kernels.WhiteKernel(1, (1e-8, 1e6)) + gp.kernels.RBF(1,
(1e-7, 1e6))
model_tlf = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=10, alpha=0.1, normalize_y=True)
model_tlf.fit(tlf_monthly_x_tr, tlf_monthly_y_tr)
params_tlf = model_tlf.kernel_.get_params()

model_tlf.score(tlf_monthly_x_tr, tlf_monthly_y_tr)

y_pred_tlf, std_tlf = model_tlf.predict(tlf_monthly_x_test,
return_std=True)
MSE_ib = ((y_pred_tlf-tlf_monthly_y_test)**2).mean()
MSE_ib

rmse = np.sqrt((y_pred_tlf-tlf_monthly_y_test)**2).mean()
rmse

tlf_monthly_x_tot = tlf_monthly['close'].to_numpy().reshape(-1, 1)
tlf_monthly_y_tot = tlf_monthly['Rentabilidad'].to_numpy().reshape(-1,
1)

y_pred_tlf_total, std_tlf_total = model_tlf.predict(tlf_monthly_x_tot,
return_std=True)
tlf_monthly['PredicciónTotal'] = tlf_monthly['Rentabilidad']
tlf_monthly.iloc[:, 2] = y_pred_tlf_total[:,0]
tlf_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(tlf_monthly.index, tlf_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(tlf_monthly.index, tlf_monthly['PrediccionTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=tlf_monthly.index,
    y1=(tlf_monthly['PrediccionTotal']-1.96*std_tlf_total),
    y2=(tlf_monthly['PrediccionTotal']+1.96*std_tlf_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Rentabilidad Mensual')
pl.ylabel('Stock Return')
pl.legend(loc="upper left")
pl.title('Telefonica Monthly Returns Prediction vs Real')
```

Real Estate Services Model

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as pl
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
import sklearn.gaussian_process as gp
```

```
from math import floor
import urllib.request
import os
from scipy.io import loadmat
import random

# # Extracción Datos
yahoo_financials = YahooFinancials('MRL.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
merlin_monthly = pd.DataFrame(data['MRL.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
merlin_monthly.head()

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(merlin_monthly.index, merlin_monthly['close'])
pl.title('Merlin Monthly Stock Price (2017-2022)')
pl.ylabel('Stock price (€)')

merlin_monthly['Rentabilidad'] = merlin_monthly['close'].pct_change(1)
merlin_monthly['Rentabilidad'] =
merlin_monthly['Rentabilidad'].apply(lambda x: x*100)
merlin_monthly = merlin_monthly.dropna()
merlin_monthly

fig = pl.figure(figsize = (15,7))
pl.xticks(rotation=90)
pl.plot(merlin_monthly.index, merlin_monthly['Rentabilidad'])
pl.title('Merlin Monthly Stock Returns (2017-2022)')
pl.ylabel('Stock returns (%)')

import random
train_n = random.sample(range(0, len(merlin_monthly['Rentabilidad'])),
45)
merlin_monthly_tr = merlin_monthly.iloc[train_n]
merlin_monthly_test =
merlin_monthly.drop(merlin_monthly.index[train_n])

print(merlin_monthly_tr.shape, merlin_monthly_test.shape)

merlin_monthly_x_tr = merlin_monthly_tr['close'].to_numpy().reshape(-
1, 1)
merlin_monthly_y_tr =
merlin_monthly_tr['Rentabilidad'].to_numpy().reshape(-1, 1)
merlin_monthly_x_test =
merlin_monthly_test['close'].to_numpy().reshape(-1, 1)
merlin_monthly_y_test =
merlin_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)


# ## GP Regression

kernel = gp.kernels.WhiteKernel(1, (1e-8, 1e6)) + gp.kernels.RBF(1,
(1e-7, 1e6))
```

```
model_merlin = gp.GaussianProcessRegressor(kernel=kernel,
n_restarts_optimizer=20, alpha=0.001, normalize_y=True)
model_merlin.fit(merlin_monthly_x_tr, merlin_monthly_y_tr)
params_merlin = model_merlin.kernel_.get_params()

model_merlin.score(merlin_monthly_x_tr, merlin_monthly_y_tr)

y_pred_merlin, std_merlin =
model_merlin.predict(merlin_monthly_x_test, return_std=True)
MSE_merlin = ((y_pred_merlin-merlin_monthly_y_test)**2).mean()
MSE_merlin

rmse = np.sqrt((y_pred_merlin-merlin_monthly_y_test)**2).mean()
rmse

merlin_monthly_x_tot = merlin_monthly['close'].to_numpy().reshape(-1,
1)
merlin_monthly_y_tot =
merlin_monthly['Rentabilidad'].to_numpy().reshape(-1, 1)

y_pred_merlin_total, std_merlin_total =
model_merlin.predict(merlin_monthly_x_tot, return_std=True)
merlin_monthly['PrediccíónTotal'] = merlin_monthly['Rentabilidad']
merlin_monthly.iloc[:, 2] = y_pred_merlin_total[:,0]
merlin_monthly

fig = pl.figure(figsize = (15,7))
pl.plot(merlin_monthly.index, merlin_monthly['Rentabilidad'], 'r-',
linewidth = 1, label=u'Real')
pl.plot(merlin_monthly.index, merlin_monthly['PrediccíónTotal'], 'b-',
linewidth = 1, label=u'Prediction')
pl.fill_between(
    x=merlin_monthly.index,
    y1=(merlin_monthly['PrediccíónTotal']-1.96*std_merlin_total),
    y2=(merlin_monthly['PrediccíónTotal']+1.96*std_merlin_total),
    color='blue',
    alpha = 0.1,
    label='Credible Interval'
)
pl.xlabel('Fecha')
pl.xticks(rotation=90)
pl.ylabel('Rentabilidad Mensual')
pl.ylabel('Stock Return')
pl.legend(loc="upper left")
pl.title('Merlin Monthly Returns Prediction vs Real')
```

Support Vector Machine Regression

```
import pandas as pd
import numpy as np
import yfinance as yf
from yahoofinancials import YahooFinancials
from matplotlib import pyplot as plt
import random
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import svm
```

```
def svm_reg(df):
    train_n = random.sample(range(0, len(df['Rentabilidad'])),
round(len(df['Rentabilidad'])*0.8))
    df_monthly_tr = df.iloc[train_n]
    df_monthly_test = df.drop(df.index[train_n])
    df_monthly_x_tr = df_monthly_tr['close'].to_numpy().reshape(-1, 1)
    df_monthly_y_tr =
df_monthly_tr['Rentabilidad'].to_numpy().reshape(-1, 1)
    df_monthly_x_test = df_monthly_test['close'].to_numpy().reshape(-
1, 1)
    df_monthly_y_test =
df_monthly_test['Rentabilidad'].to_numpy().reshape(-1, 1)
    regr = svm.SVR()
    regr.fit(df_monthly_x_tr, df_monthly_y_tr)
    y_pred = regr.predict(df_monthly_x_test)
    rmse = np.sqrt((y_pred-df_monthly_y_test)**2).mean()
    r2 = r2_score(df_monthly_y_test, y_pred)
    return r2, rmse

yahoo_financials = YahooFinancials('IBE.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['IBE.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('ACS.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-06-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['ACS.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('ITX.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-06-
01',
```

```
                                                    end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['ITX.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('AENA.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-06-
01',
                                                    end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['AENA.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('BBVA.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                    end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['BBVA.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('TEF.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                    end_date='2022-02-
15',

time_interval='monthly')
```

```
df_monthly = pd.DataFrame(data['TEF.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)

yahoo_financials = YahooFinancials('MRL.MC')
data = yahoo_financials.get_historical_price_data(start_date='2017-01-
01',
                                                  end_date='2022-02-
15',

time_interval='monthly')
df_monthly = pd.DataFrame(data['MRL.MC']['prices']).drop(columns =
['date', 'high', 'low', 'open', 'volume',
'adjclose']).set_index('formatted_date')
df_monthly['Rentabilidad'] = df_monthly['close'].pct_change(1)
df_monthly['Rentabilidad'] = df_monthly['Rentabilidad'].apply(lambda
x: x*100)
df_monthly = df_monthly.dropna()

r2, rmse = svm_reg(df_monthly)
print(r2, rmse)
```