Predicting ECB Interest Rates with ML

An in-depth analysis of techniques to forecast ECB Interest Rates.

Hephaestus Applied Artificial Intelligence Association

Authors:

Member	Role
Filippo A. Ronzino	Head
Elisa Tofanelli	Member
Samuele Straccialini	Member
Micaela Contini	Member
Stefano Di Filippo	Member
Francesco Sassi	Member
Antonio Honsell	Member
Marit Huenlein	Member
Camilla Trudda	Member
Gergana Tagareva	Member

Milan, February 12, 2024

Contents

1 | Introduction

1.1 | Motivations

In economics, the prediction of data has always played a pivotal role: interest rates, unemployment, and inflation are only some of the hundreds of intertwined rates that affect everyone's lives, directly or indirectly. Indeed, whole areas of finance focus on analyzing past data and events to anticipate future developments. This involves employing complex mathematical formulas, conducting in-depth analyses of financial statements and ratios, staying attentive to the news, and searching for patterns in the past that enable us to foresee the future. A single or general "interest rate" does not exist: each lender can offer a different one, more or less convenient, without affecting more than a few people; the ones everyone is trying to predict are those set by Central Banks. These are financial institutions that manage the currency of a country or group of countries and control the monetary supply. This means that a Central Bank has the power to control the economy through monetary policies - such as setting interest rates. In this study, we are trying to predict the interest rates set by the European Central Bank (**ECB**, in short) that affect directly the eurozone and indirectly the world's economy. The ECB sets different interest rates: the **MLR** (Marginal Lending Rate), the **DER** (Deposit Facility Rate), and the **MRR** (Marginal Refinancing Rate), the most important and the one we are trying to predict, which represents the rate at which eurozone commercial banks can borrow funds from the ECB. Those banks then lend money to the public at a rate slightly higher than the MRR: from the person who's asking for a loan to buy his first house to the entrepreneur who is seeking funds for his start-up, everyone is eventually affected by the interest rates set by the ECB. To predict the MRR we are using Linear Regression, Support Vector Regression, and ARIMA, assuming that the interest rate is set based on quantitative economic indicators such as inflation, GDP growth, or other factors described in the following sections. Being able to predict this value accurately and consistently would imply being two steps ahead of everyone who is not able to do so, exploiting this knowledge as one wishes to obtain anything ranging from wealth to power and influence.

1.2 | Regression Problem

The aim of the project is therefore to translate a financial problem into a regression problem and more generally into other statistical learning models which provide the foundational framework to the field of machine learning, integrating principles from statistics and functional analysis. The core objective of statistical learning is to identify an optimal function that captures the systematic relationships between predictors and responses. This function is pivotal for making predictions and drawing inferences. The versatility of statistical learning theory is evident in its applications to regression modeling and data classification. In regression scenarios, nominally the ones we are interested in, responses are quantitative values within a continuous range, while classification deals with qualitative values from a discrete set of labels. More specifically for the first case, given $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$ the dataset we will consider has the form

$$\mathcal{D} := \{(x_1, y_1), ..., (x_p, y_p)\}$$

which is indeed a matrix containing predictors $(x_{11}, ..., x_{1p})...(x_{n1}, ..., x_{np})$ for $i = 1, ..., n$. Hence in the most general regression scope, where θ parameter vector[1] and ϵ statistical noise, we want to determine the unknown function f [2] that explains the covariates:

$$Y_i = f_\theta(x_i) + \epsilon_i \tag{1.1}$$

The latter is in fact called *noisy observation*, which is explained by the fact that ϵ is an i.i.d. random variable that describes measurement/observation noise and potentially unmodeled processes. In the end, our task is to find a function that not only models the training data but generalizes well to predicting function values at input locations that are not part of the training data. Finding a regression function requires solving a variety of problems, including the following:

- **Choice of the model (type) and the parametrization of the regression:** Given a dataset, what function classes (e.g., polynomials) are good candidates for modeling the data, and what particular parametrization (e.g., degree of the polynomial) should we choose?

[1] In the linear regression the vector θ will correspond to the coefficients $\omega_1, ..., \omega_p$ that we want to find

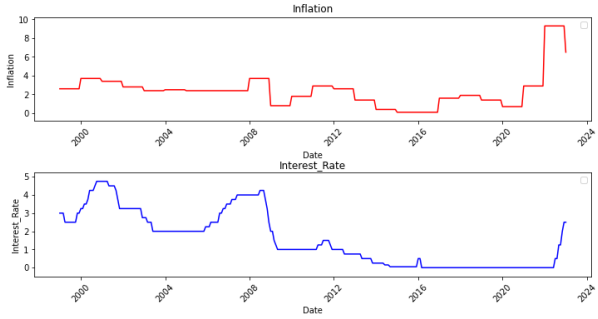
[2] If the form of the function f is not specified beforehand, then we speak of nonparametric regression: we then use the observations to determine a suitable type of function and approximation methods, including Fourier series, wavelets, spline functions, and neural networks.

- **Finding good parameters:** Having chosen a model of the regression function, how do we find good model parameters? Here, we will need to look at different loss/objective functions (they determine what a "good" fit is) and optimization algorithms that allow us to minimize this loss.
- **Overfitting and model selection:** Overfitting is a problem when the regression function fits the training data "too well" but does not generalize to unseen test data

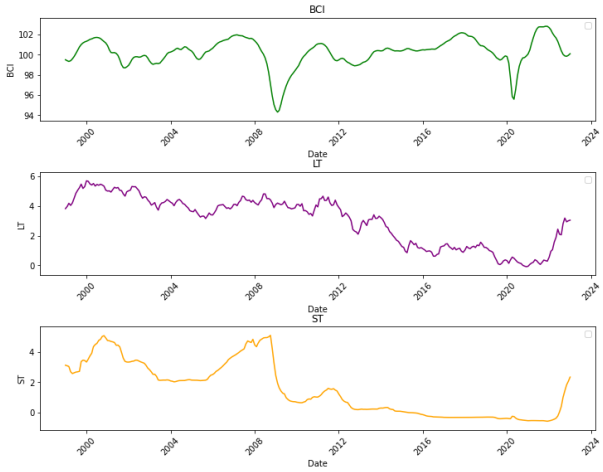
Despite variations in the approach to unveil the function f, such as the choice of the loss function, commonalities exist across both regression and classification problems but the upcoming chapters delve into the regression problem which is at the essential core of our problem.

1.3 | Dataset

For our dataset, we obtained the data directly from the ECB website [1], specifically selecting the list of monthly values of the key economic indicators we were interested in: Interest Rate, Inflation, Business Confidence Index (BCI)[3] [7](which was later removed from the multiple linear regression due to insufficient correlation with the Interest Rate), Long Term Interest Rate (LT), and Short Term Interest Rate (ST). Inflation for the Euro-Zone will moreover be considered for the last models as a feature[4].
To ensure data consistency, we focused on the period from January 1999 to January 2023 for the first two models and subsequently added the January-September 2023 period for ARIMA in order to get accurate forecasting. During the data preprocessing phase, we addressed missing values by filling empty columns. To achieve this, we imputed the value of the missing month with that of the preceding one, ensuring a continuous and coherent dataset for analysis.



(a) Inflation and Interest Rate



(b) LT, ST and BCI

Figure 1.1: Data Visualization of the features

[3]BCI provides information on future developments, based upon opinion surveys on developments in production, orders, and stocks of finished goods in the industry sector. It can be used to monitor output growth and to anticipate turning points in economic activity. It is averaged at 100: so a value above 100 indicates an increased confidence in near future business performance, and numbers below 100 pessimism towards future performance.
[4]We will use the terms variables, covariates, and features interchangeably in this report.

2 | Linear Regression

2.1 | Mathematical Framework

Our analysis starts with the simplest possible regression model, namely linear regression. In order to formulate a learning problem mathematically, we need to define two things: a model and a loss function. The **model**, or architecture, defines the set of possible hypotheses or functions that compute predictions from the inputs. In the case of linear regression, the model simply consists of linear functions. Recall that a linear function of D inputs is parameterized in terms of D coefficients, which we'll call the weights, and an intercept term, which we'll call the bias. Mathematically (we refer to the notation reported in [4]), this is written as:

$$y = \sum_j \omega_j x_j + b \tag{2.1}$$

Clearly, some of the linear fits are better than others. In order to quantify how good the fit is, we define a **loss function**. This is a function $\mathcal{L}(y, t)$ which says how far off the prediction y is from the target t (this becomes evident in 2.2). In linear regression, we use squared error, defined as:

$$\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2 \tag{2.2}$$

In general, the value $(y - t)$ is known as the **residual**, and we want the residuals to be as close to zero as possible. When we combine our model and loss function, we get an **optimization problem**, where try to minimize a cost function with respect to the model parameters (i.e. the weights and bias). The cost function is simply the loss, averaged over all the training examples:

$$\mathcal{E}(\omega_1, ..., \omega_D, b) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, t_i) \tag{2.3}$$

$$= \frac{1}{2N} \sum_{i=1}^{N} \Big(\sum_j w_j x_{ij} + b - t_i \Big)^2 \tag{2.4}$$

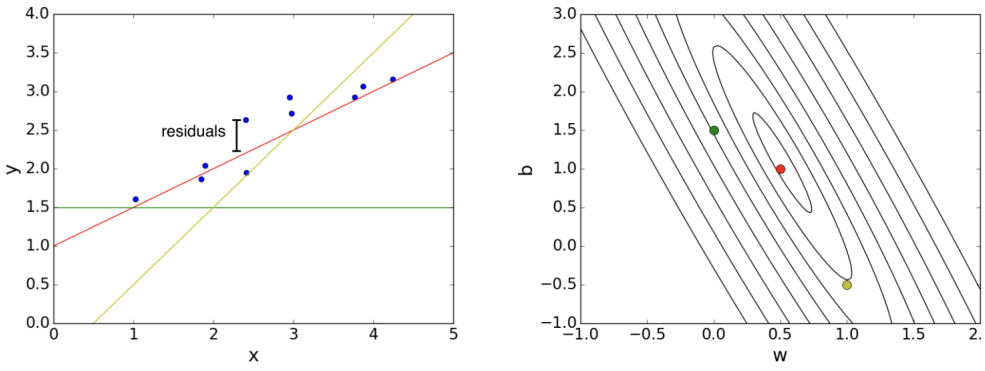


Figure 2.1: Residuals and Contour plot of least-squares cost function for regression (example in [4])

Hence, our goal is to choose $\omega_1, ..., \omega_D$ and b to minimize $\mathcal{E}$. In other terms, calling $\mathbf{x} = (x_1, ..., x_D)$ the realizations of the random vector X then, what we are really assuming is that $\mathbb{E}(Y|X = \mathbf{x}) = \sum_j \omega_j x_j$ and $\text{Var}(Y|X = \mathbf{x}) = \sigma^2$ which means that the model has D+1 real-valued parameters, which we can combine into the parameter vector $\theta = (\omega_1, ..., \omega_D, \sigma^2)$. Finally defining $\epsilon := Y - \sum_j \omega_j x_j$ our noise term is normally distributed with mean 0 and variance σ^2. It is clear to see that the **MLE**[5] for the parameter

[5]The proof is done by direct computations in the simple case of three-dimensional θ parameter, i.e. considering the log-likelihood of the model and setting to zero the partial derivatives, while for the more general case it relies on the notion of projection matrix which is beyond the scope of this project, see [2]

vector θ, calling with X the design matrix, i.e. $X = \begin{bmatrix} x_{11} & \dots & x_{1D} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nD} \end{bmatrix}$ is:

$$\hat{\omega} = (X^T X)^{-1} X^T Y, \quad \hat{\sigma^2} = \frac{\|Y - X\hat{\omega}\|^2}{n}$$

In the end, the linear function in 2.1 can be more elegantly rewritten as $f(x) = \langle \omega, \mathbf{x} \rangle + b$[6] and so, the optimal regression function is given by solving the minimization[7] problem $\min \sum_i (y_i - \hat{y}_i)^2$, hence to find the best fit, we minimize the sum of squared errors, i.e. the loss function in this case.

2.2 | Implementation

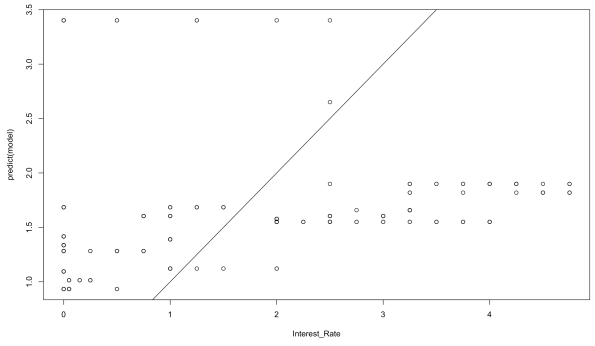
From a practical point of view, we utilize R software to conduct our analysis. We use the dataset mentioned in Section 1.3; it consists in 289 observations (one per month) from January 1999 to January 2023. The columns are the following:

- **Interest_Rate** - Observed Interest Rate value
- **Inflation** - Observed Inflation value
- **BCI** - Business Confidence Index
- **ST** - Short Term interest rate
- **LT** - Long Term interest rate

Our initial strategy involved implementing a simple linear regression model with inflation as the independent variable, aiming to capture the relationship expressed as $Y_i = \beta x_i + \epsilon$, where Y_i represents the interest rate, x_i denotes inflation, β signifies the slope and ϵ stands for the error term. The choice of inflation as a co-variate derives from the fact that we anticipated this to be the main factor influencing interest rates. Fig.2.2 illustrates that while a minor correlation between the two may be found, this approach does not yield optimal results. Indeed, an optimal plot of the predicted values against the actual ones would imply that the two lie along the $y = x$ line, which was not our case.

```
Call:
lm(formula = Interest_Rate ~ Inflation)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4020 -1.0324 -0.3897  0.9493  2.9310

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.90680    0.13863   6.541 2.79e-10 ***
Inflation    0.26830    0.04683   5.730 2.54e-08 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.427 on 287 degrees of freedom
Multiple R-squared:  0.1026,    Adjusted R-squared:  0.09952
F-statistic: 32.83 on 1 and 287 DF,  p-value: 2.543e-08
```

(a) R output



(b) Scatterplot of predictions against real interest rate

Figure 2.2: Simple linear regression for `Interest_Rate` $\sim$ `Inflation`.

The inefficiency of the model is also reflected by the computed R-squared[8] value of approximately 0.09952, which indicates that inflation accounts for only a small fraction of the variability observed in interest rates and suggests the presence of other influential variables that have not been accounted for.

In order to address this limitation and get a more accurate result, we adopted a second strategy using all explanatory variables in a multiple linear regression. We loaded the necessary dataset and proceeded to build the model. Once the model was constructed, we examined the summary statistics to gain insights

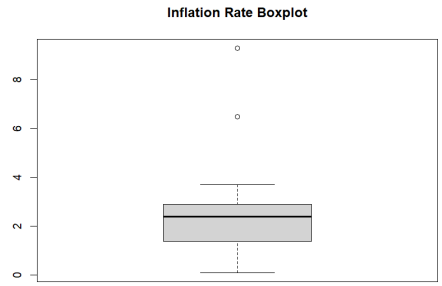
[6] $\langle \cdot, \cdot \rangle$ denotes the inner product in the Euclidean space.
[7] A regularization term can be introduced, see LASSO or Ridge Regression
[8] R-squared determines the proportion of variance in the dependent variable explained by the independent variable.

into the coefficients, significance levels, and overall goodness-of-fit of the model. These insights helped asses the variables' impacts and their significance in predicting interest rate.

The second approach permits us to identify a better correlation between our independent variables and the interest rate: the inclusion of additional explanatory variables led to more robust and statistically significant relationships with the dependent variable. Upon examining the coefficients shown in Fig.2.3a, we observed the following key findings. In the first place, the coefficient estimates for **Inflation** and **ST** were found to be highly significant, both with a p-value[9] smaller than 0.001. On the contrary, the coefficient for **BCI** showed that it was not statistically significant enough ($p = 0.660833$), suggesting its negligible role in explaining interest rate variability in the model. Consequently, we opted to remove it from our analysis[10]. To further enhance model accuracy, we also identified and removed outliers in **LT** and **ST**; Fig.2.3b provides a visual representation of the aforementioned data points.

```
Call:
lm(formula = Interest_Rate ~ Inflation + BCI + LT + ST)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84477 -0.09934 -0.00803  0.12655  0.62406

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.130638   0.881987  -0.148 0.882355
Inflation    0.025577   0.007082   3.612 0.000359 ***
BCI          0.003846   0.008756   0.439 0.660833
LT          -0.017303   0.014229  -1.216 0.224987
ST           0.847611   0.013242  64.007  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2005 on 284 degrees of freedom
Multiple R-squared:  0.9825,    Adjusted R-squared:  0.9822
F-statistic:  3978 on 4 and 284 DF,  p-value: < 2.2e-16
```

(a) R output



(b) Boxplot of Interest Rate showing outliers

Figure 2.3: Multiple linear regression including all variables

2.3 | Results

Our refined model, excluding **BCI** and outliers, yielded remarkable improvements in explaining interest rate variability (all results are summarized in Fig.2.4). Notably, the adjusted R-squared value of 0.9846 indicates that approximately 98.46% of the variance in interest rates can be explained by the model's variables.
In particular, R output showed that we had enough statistically significant evidence to consider all variables useful – **Inflation**, **LT** and **ST** – in predicting interest rate values (Fig.2.4a). We further assessed the model's performance through visual diagnostics, plotting predicted values against true values of the dependent variable (Fig.2.4b). In conclusion we checked linear regression assumptions: QQ-plot in Fig.2.4c showed that residuals were mostly normally distributed. Also, a scatterplot of residuals against fitted model values suggested the homoscedasticity [11] assumption was fairly respected (Fig.2.4d), confirming that the model's predictions were consistent with the underlying statistical assumptions.

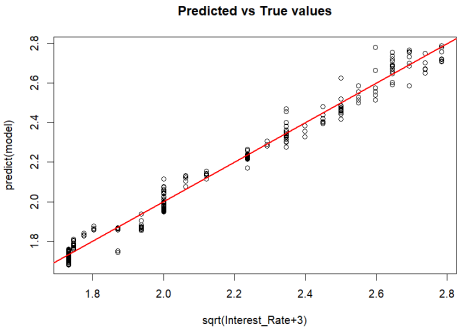
[9] p-value, $PR(> |t|)$ in Fig.2.3a, represents the likelihood, within a statistical model, that the observed results or more extreme outcomes would occur if the null hypothesis were true.

[10] This procedure is known as step-down/backward selection and consists in iteratively eliminating from the analysis the covariate with the highest p-value (considering ones above a certain threshold) to get a simpler model. Although the resulting model is not guaranteed to be the best one possible, this selection procedure is valuable for its simple and straightforward implementation and interpretation.

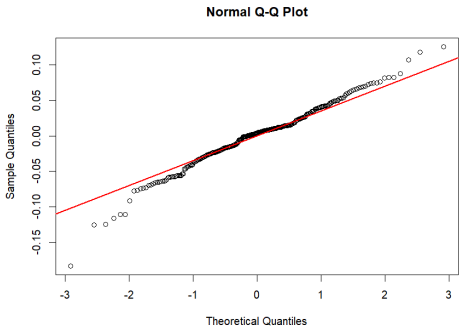
[11] Homoscedasticity means errors have constant variance. Normally distributed errors with constant variance is indeed a fundamental assumption of linear regression.

```
Call:
lm(formula = sqrt(Interest_Rate + 3) ~ Inflation + LT + ST, data =
no_outliers_data)

Residuals:
      Min        1Q    Median        3Q       Max
-0.183591 -0.023501  0.004125  0.023806  0.125745

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.762476   0.007849 224.549  < 2e-16 ***
Inflation   0.018275   0.003514   5.201 3.91e-07 ***
LT          0.012476   0.002978   4.190 3.78e-05 ***
ST          0.175331   0.003099  56.576  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04306 on 272 degrees of freedom
Multiple R-squared:  0.9848,    Adjusted R-squared:  0.9846
F-statistic:  5876 on 3 and 272 DF,  p-value: < 2.2e-16
```

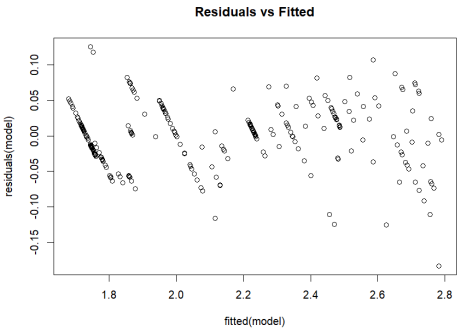
(a) R output



(b) Predicted against true interest rate



(c) Normality of residuals



(d) Homoscedasticity of residuals

Figure 2.4: Final multiple linear regression

3 | Support Vector Regression

3.1 | Mathematical Framework

We extend our purpose to capture the possible non-linear relations between the covariates and the ECB Interest Rate. For this purpose, we introduce the notion of Support Vector Machine, from which we will consider the regression part, i.e. the Support Vector Regression[12]. Indeed, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces but can also be used for regression tasks, where the objective becomes ϵ-sensitive.

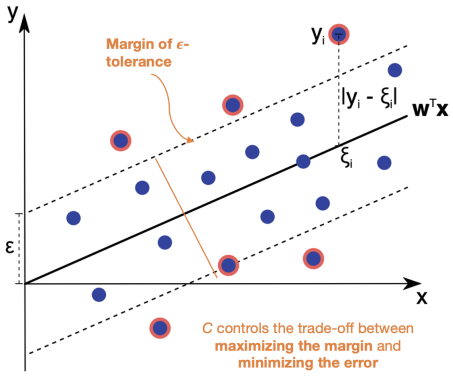


Figure 3.1: Support Vector Regression in dimension 2

A support vector machine constructs a hyperplane[13] or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the **largest distance** to the **nearest training-data point** of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier. A lower generalization error means that the implementer is less likely to experience overfitting. Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that **dot products** of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a **kernel function** $k(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose *dot product with a vector in that space is constant*, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors x_i that occur in the data base. With this choice of a hyperplane, the points $\mathbf{x}$ in the feature space that are mapped into the hyperplane are defined by the relation $\sum_i \alpha_i k(x_i, x) = const.$ In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Hence, our general regression problem (see section 1) to find the function $f(\mathbf{x})$ now translates into a convex optimization one, letting x_i a training sample with target value y_i[14]:

$$\min \frac{1}{2}\|\omega\|^2 \tag{3.1}$$

$$s.t. \quad y_i - \langle \omega, x_i \rangle - b \leq \epsilon \tag{3.2}$$

$$\langle \omega, x_i \rangle - y_i + b \leq \epsilon \tag{3.3}$$

Now, we start from the case in which the function f can be linearly parametrized as in the linear regression model i.e. $f(\mathbf{x}) = \langle \omega, \mathbf{x} \rangle + b$. In this situation, it is easy to see that only the point outside the ϵ-region

[12] A version of SVM for regression was proposed in 1996 by Vladimir N. Vapnik, Harris Drucker, Christopher J. C. Burges, Linda Kaufman and Alexander J. Smola

[13] Please refer to Appendix A to see the details

[14] References in [5]

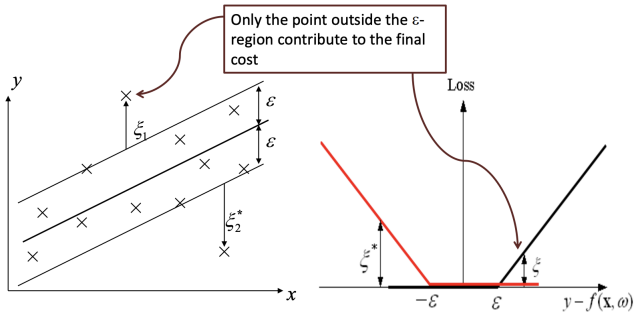


Figure 3.2: Loss function

contributing to the final cost, and hence the ϵ-insensitive loss function is the following:

$$\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})| \tag{3.4}$$

This loss function is ideal when small amounts of error are acceptable, a condition often referred to as 'soft-margin'. In ϵ-insensitive loss function, any points within some selected range ϵ are considered to have no error at all, which means the ϵ-insensitive loss function can be represented as:

$$\mathcal{L}_\epsilon(y) = \begin{cases} 0 & |f(\mathbf{x}) - y| < \epsilon \\ |f(\mathbf{x}) - y| - \epsilon & otherwise \end{cases} \tag{3.5}$$

This error-free margin makes the loss function an ideal candidate for support vector regression. Now, the minimization problem in 3.1 is not always feasible of course, hence slack variables (ξ) are usually added into the above to allow for errors and to allow approximation. In this case the problem reads:

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_i (\xi_i^- + \xi_i^+) \tag{3.6}$$

$$s.t. \tag{3.7}$$

$$\begin{aligned} &y_i - \langle \omega, x_i \rangle - b \leq \epsilon + \xi_i^- \\ &\langle \omega, x_i \rangle - y_i + b \leq \epsilon + \xi_i^+ \\ &\xi_i^-, \xi_i^+ \geq 0 \end{aligned} \tag{3.8}$$

where $C > 0$ is a pre-specified constant value that determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated, and the ξ_i^-, ξ_i^+ are slack variables representing upper and lower constraints on the outputs of the system.
Finally, to deal with the situation of non-linear support vector regression, we introduce what is called the *kernel approach* or *kernel trick* to allow us to deal with the mapping $\varphi : X \to F$, with X input space and F another standard space where standard SV linear regression performs. This kernel trick avoids the explicit mapping that is needed to get linear learning algorithms to learn a non-linear function or decision boundary. In this sense, recalling that $k : X \times X \to \mathbb{R}$ then:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_F \tag{3.9}$$

The key restriction is that $\langle \cdot, \cdot \rangle_F$ must be a proper inner product. On the other hand, an explicit representation for φ is not necessary as long as F is an inner product space.[15] With this trick the regression hyperplane is simply:

$$f(\mathbf{x}) = \langle \omega, \varphi(\mathbf{x}) \rangle + b \tag{3.10}$$

and hence the minimization problem becomes a variation of the previous one:

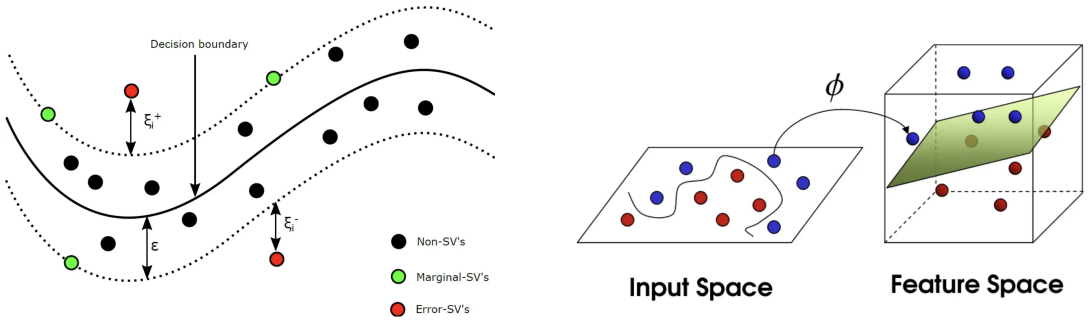
$$\min \frac{1}{2}\|\omega\|^2 + C\sum_i (\xi_i^- + \xi_i^+) \tag{3.11}$$

$$s.t. \tag{3.12}$$

$$\begin{aligned} &y_i - \langle \omega, \varphi(x_i) \rangle - b \leq \epsilon + \xi_i^- \\ &\langle \omega, \varphi(x_i) \rangle - y_i + b \leq \epsilon + \xi_i^+ \\ &\xi_i^-, \xi_i^+ \geq 0 \end{aligned} \tag{3.13}$$

[15] An alternative to that requirement is that X is equipped with a suitable measure ensuring that k satisfies Mercer's Conditions, see [6] for details.

Before proceeding with the implementation we'll see a few qualitative remarks[16] just to fix the idea of kernel trick. Consider our task of predicting the ECB Interest Rate from a certain covariate that we give as an input, say we consider the Inflation Rate. At the beginning what we did was try to find a fitting *linear* function of x to the training data. What if y can be more accurately represented by a *non-linear* function of x? In this case, we need a more expressive family of models than linear models, which are the ones introduced in Section 1 when dealing with the regression problem. Hence, SVR aims to find a function that maps input features to corresponding output values, making it a regression task. Unlike traditional linear regression, SVR allows for the identification of non-linear relationships by introducing a mapping into a higher-dimensional space through the use of kernel functions, minimizing the error between the predicted and actual output values while maximizing the margin around the regression line.



(a) Support Vectors, ϵ margin and slacks

(b) Kernel functions

While linear SVR uses linear kernel function and aims to find a linear hyperplane that best fits the data, non-linear SVR employs various kernel functions (e.g., polynomial, Gaussian radial basis function, etc.) to capture complex relationships between variables.
In order to train a support vector regression and optimize our objective function, we would have to perform operations with the higher dimensional vectors in the transformed feature space. In real applications, there might be many features in the data, and applying transformations that involve many polynomial combinations of these features will lead to extremely high and impractical computational costs. The kernel trick provides a solution to this problem. The "trick" is that kernel methods represent the data only through a set of pairwise similarity comparisons between the original data observations x (with the original coordinates in the lower dimensional space), instead of explicitly applying the transformations $\varphi(x)$ and representing the data by these transformed coordinates in the higher dimensional feature space. Our kernel function accepts inputs in the original lower dimensional space and returns the dot product of the transformed vectors in the higher dimensional space. The ultimate benefit of the kernel trick is that the objective function we are optimizing to fit the higher dimensional decision boundary only includes the dot product of the transformed feature vectors. Therefore, we can just substitute these dot product terms with the kernel function, and we do not even need the explicit formula for $\varphi(x)$.

3.2 | Implementation

For practical reasons, we switched to Python and implemented our SVR model using the scikit-learn 1.4.0 library. The first step consists in loading the data set from a CSV file `output_file.csv` which contains the data described in Section 1, which has been preprocessed in the appropriate format to be fed to the scikit-learn methods. This includes:

- **Feature Extraction.** The four economic indicators, Inflation, BCI, LT, and ST, are extracted from the dataset and designated as independent variables (X). The target variable, Interest Rate, is identified as the dependent variable (y); the justification of our choice of the independent variables derives from economic considerations which imply that there is a significant correlation between each one of these and the Interest rate.
- **Data Splitting.** In order to evaluate the model's performance effectively, the dataset was split into $K = 5$ folds using K-Fold cross-validation. This technique involves dividing the data into multiple subsets, training the model on each subset, and evaluating its performance on the remaining subsets. This allows us to determine whether the model is overfitting the data in learning the relationship between the features and the target variable.

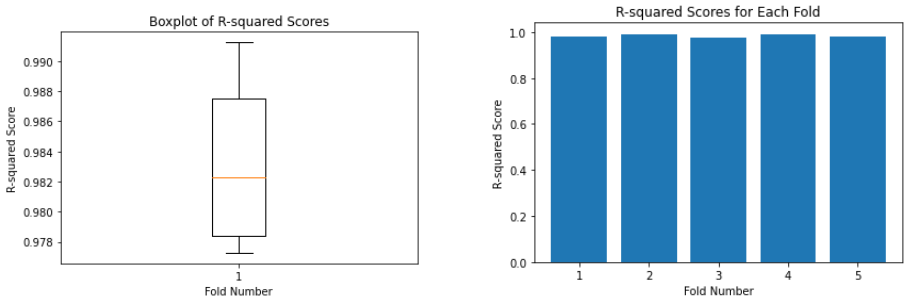
[16] References in [8].

- **Feature Scaling.** Standardization was applied to the features (X) to ensure that they were on a similar scale. This is crucial for SVR models, as they are sensitive to the scale of the input data. Standardization involves subtracting the mean and dividing by the standard deviation of each feature.

The SVR model was trained using various kernel functions, including the linear kernel. The radial basis function (RBF) was chosen because it provided the best results in terms of R-squared score and mean squared error (MSE). Indeed, in our case, the RBF kernel proved to be flexible and capable of handling nonlinear relationships between features and the target variable. After extensive experimenting, including a run of five-fold cross-validation using the corresponding training data, to avoid overfitting we set the model parameters, gamma and epsilon, to 0.1 and 0.05, respectively. The latter choice is consistent with the goal of making the training loss function stricter. The remaining parameters were the default parameters since there was no clear motivation for changing them. We have also carried out various experiments changing the learning subset and concluded that the dataset seems rather robust in training the SVR model.

3.3 | Results

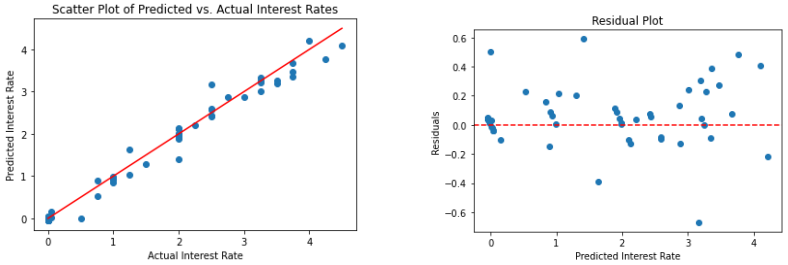
To assess the performance of the trained SVR models, two metrics are employed as in the previous Linear Regression: R-squared score, mean squared error (MSE), and mean absolute error (MAE). R-squared measures the proportion of variance in the target variable that can be explained by the model. A higher (closer to 1) R-squared score indicates a better fit of the model. MSE quantifies the average squared difference between the predicted and actual values of the target variable, hence a lower (closer to 0) MSE indicates better model performance. The results are satisfying: over the five folds the R^2 has a minimum value of 0.977, mean of 0.983, and standard deviation of 0.005, while the MSE maximum is 0.05 and MAE is 0.137 which, indeed, suggests that the model is consistently accurate across all folds.



(a) Boxplot of R-squared values **(b)** Histogram for each fold

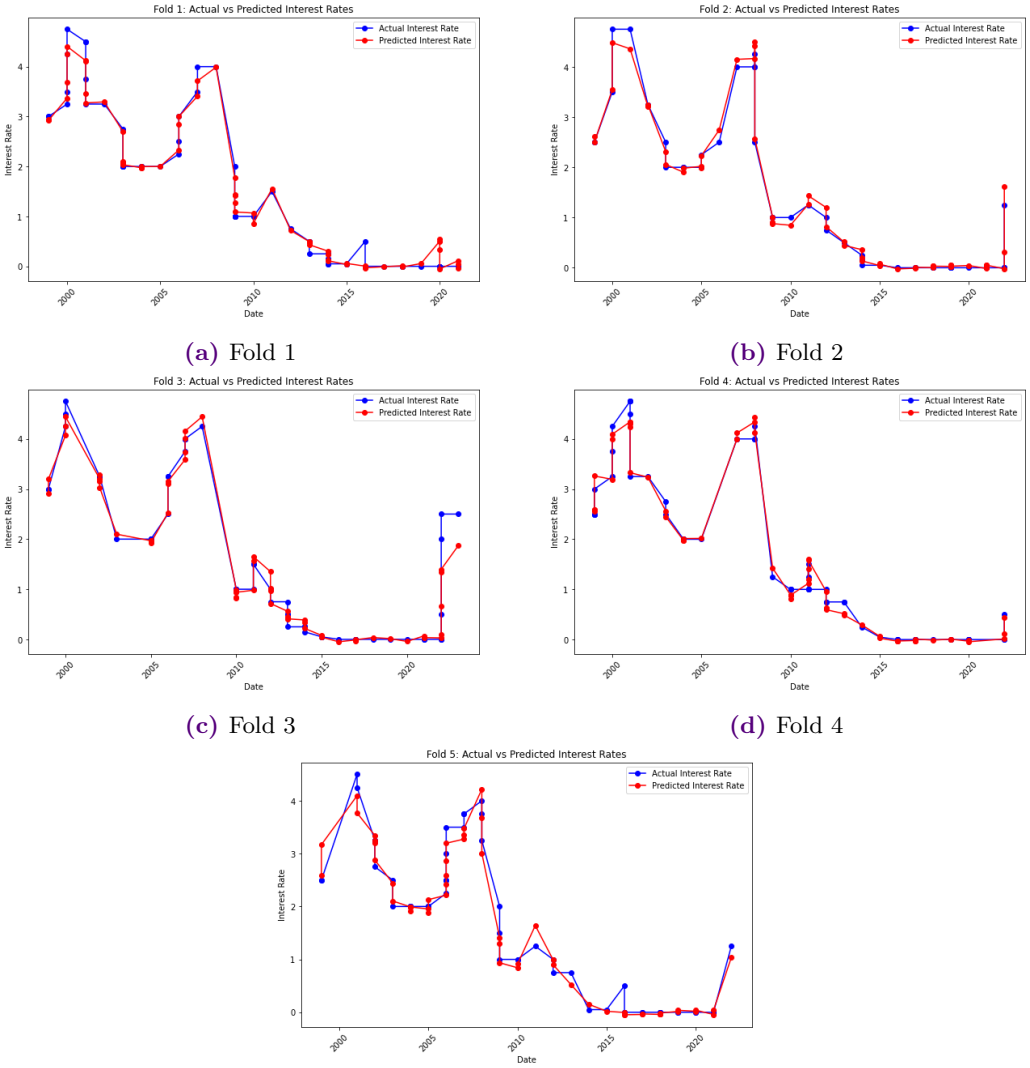
Figure 3.4: R-squared results for SVR

Moreover, it is even more evident from the time-series plot for each fold that the model makes relevant predictions compared to actual values. Indeed, representing in blue the actual data points for the Interest Rate and in red the predicted ones we see a nice alignment of the two (see figure 3.6).
Another remarkable alignment on the first quadrant bisector, i.e. the *pred* = *actual* line, is visible in the scatter plot of the SVR predicted values and actual Interest Rates:



(a) Plot predicted against actual **(b)** Residual Plot

Figure 3.5: Analysis of the residuals



(a) Fold 1

(b) Fold 2

(c) Fold 3

(d) Fold 4

(e) Fold 5

Figure 3.6: Plot actual vs predicted values for each fold

Looking at the right plot of figure 3.5, it is clear that residuals are randomly scattered around the horizontal axis and there is no clear pattern or trend. This randomness indicates that the model is capturing the underlying patterns in the data well. In addition, the mean is close to zero which excludes a possible consistent deviation that may indicate a bias in the model. The learning curve analysis provides valuable insights into the performance of the SVR model as the training set size increases. The figure below (3.7) illustrates the learning curves for key metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2), across the five different folds. Notably, an evident pattern emerges, showcasing a positive trajectory in model performance with the expansion of the training set:

- There is a discernible decrease in both MSE and MAE as the training set size grows. This decline signifies an improvement in the accuracy of the SVR model's predictions, with the model achieving a closer match to the actual values. The diminishing trend in these error metrics is indicative of the model's ability to generalize well to new data points, as the increased sample size allows for a more robust understanding of underlying patterns in the dataset.
- R-squared values exhibit a consistent increase throughout the learning curve. The rising R^2 values imply an enhancement in the model's explanatory power, signifying that a larger proportion of the variance in the target variable is being captured by the SVR model. This aligns with our objective of constructing a model that not only predicts accurately but also comprehensively explains the variability in the interest rates based on the selected features.

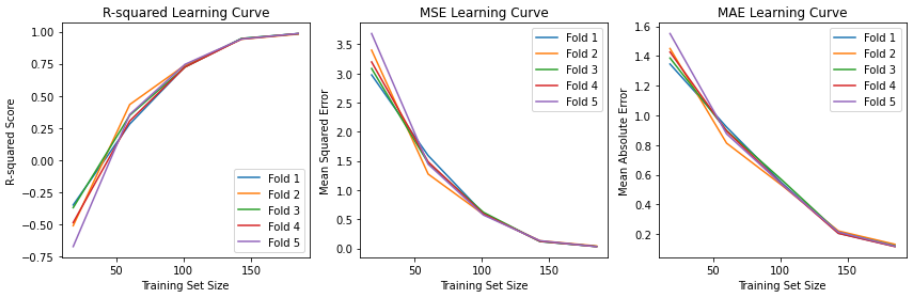


Figure 3.7: Learning Curves for metrics R-squared, MSE and MAE for each fold

While the current analysis reveals promising results, there are avenues for further refinement and enhancement of the Support Vector Regression model. Consideration should be given to exploring additional hyperparameter tuning to optimize the model's performance further. Adjustments in the choice of kernel functions, regularization parameters, and kernel coefficients could be examined to ascertain whether different configurations yield superior outcomes.

4 | AutoRegressive Integrated Moving Average

4.1 | Mathematical Framework

Recalling that the scope of the project is to give meaningful predictions about ECB Interest Rates, it is clear that the kind of data we are using are series of data points indexed in time order, hence the choice of a time series. Models for time series, included in the statistical learning field of study, use data that can have many forms and represent different stochastic processes. When modeling variations the two main questions that arise are *"How does what happened today affect what will happen tomorrow?"* and *"What is the economic cycle through periods of expansion and recession?"*; in the first case time-domain approach is used, i.e. the investigation of lagged relationships, while in the second frequency-domain approach is preferred, i.e. the investigation of cycles. For simplicity, we will deal with the univariate, linear, and discrete case of time series. In general, a time series is affected by four components[17]:

- **trend**: the general tendency of a time series to increase, decrease, or stagnate over a long period
- **seasonal**: fluctuations within a year during the season, usually caused by climate and weather conditions, customs, traditional habits, etc...
- **cyclical**: medium-term changes caused by circumstances, which repeat in cycles. The duration of a cycle extends over a longer period of time
- **irregular**: unpredictable influences, which are not regular and also do not repeat in a particular pattern. These variations are caused by incidences such as war, strike, earthquake, flood, revolution, etc...[18]

Considering the effects of these four components, two different types of models are generally used for a time series:

- **additive**: assuming these four components are independent of each other

$$Y(t) = T(t) + S(t) + C(t) + I(t) \tag{4.1}$$

- **multiplicative**: assuming these four components are not necessarily independent and they can affect one another

$$Y(t) = T(t) \cdot S(t) \cdot C(t) \cdot I(t) \tag{4.2}$$

ARIMA (Auto-Regressive Integrated Moving Average) is a model conceptually consisting of three components:

- **AR**: the autoregressive part. Autoregression is a model that regresses a variable on its past values, for a number of lags.
- **I**: the differencing of the time series in order to make it stationary.
- **MA** the moving average part. The moving average model regresses a variable on current and lagged error terms.

An ARIMA model is specified by 3 parameters: p is the order of the AR model, hence the number of included lags; d is the degree of differencing, that is, the number of times that the data have been substituted with first-order differences; q is the order of moving average model, similarly to p.
Thus, in the model, the **AR** component counts for the idea that the current value of the series, X_t, can be explained as a linear combination of p past values together with a random error w_t, whereas the **MA** component is conceptually a linear regression of the current value of the series against current and previous white noise error terms or random shocks. Combining the AR and the MA models, without first differencing the data, is what is referred to as the family of ARMA models. A time series $\{X_t : t \in \mathbb{Z}\}$ is ARMA(p, q) if we can write

$$X_t = w_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j w_{t-j} \tag{4.3}$$

[17] From now on we will refer to [9] to describe the mathematical setting.
[18] There is no defined statistical technique for measuring random fluctuations in a time series.

where $w_t \sim \text{wn}(0, \sigma_w^2)$. The stochastic process defined by this equation is stationary. This means that we cannot apply an ARMA model to a non-stationary series, which hinders its applicability a lot; for example, many time series are composed of both a non-stationary trend and a zero-mean stationary process: $X_t = \mu_t + Y_t$. To address this problem, we need ways to make the data stationary, and then apply ARIMA to the transformed observations. If the root cause of non-stationarity is the presence of unit roots in the process, the data can be made stationary through differencing. This means that we difference the series, i.e. we subtract from each observation its first-lag value, as many times as unit roots there are. Applying an ARMA(p, q) on a series which has been differeced d times is effectively what is referred to as an ARIMA(p, d, q).

Now that we have summarized the concept of ARIMA, we briefly describe the Box-Jenkins methodology ([3]), an approach designed to find a good forecasting model among ARIMA(p, d, q). It consists of three steps:

1. **Model identification**: Checking stationarity[19] and seasonality, performing differencing if necessary, and choosing model specification ARIMA(p, d, q). In particular, the choice of p and q can be guided by the observation of the autocorrelation and partial autocorrelation functions of the series.

2. **Parameter estimation**: Estimating the selected ARIMA model using maximum likelihood estimation or non-linear least-squares estimation.

3. **Diagnostic checks**: Checking the white noise assumption of residuals, the significance of model coefficients, and information criteria. If necessary, go back to step 1.

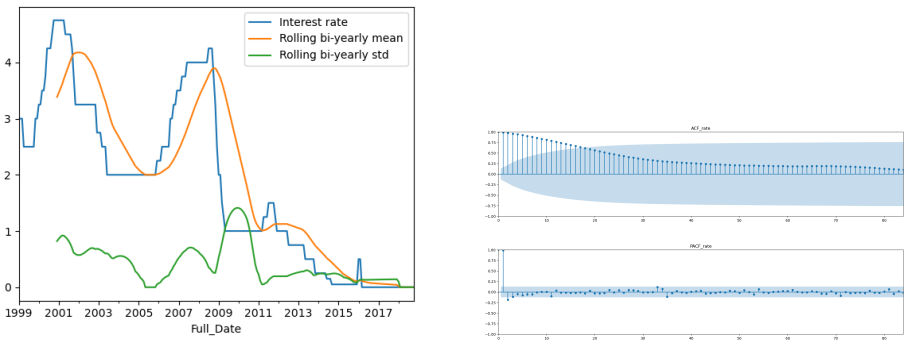
Finally, we shortly discuss the seasonal ARIMA model, or SARIMA, an extension of the just presented ARIMA to model even seasonal data. This model includes three additional components which are basically the same ones we have seen for non-seasonal ARIMA, but with seasonal backshifts, that is, with lags being multiples of the seasonal period. This means that SARIMA is specified by 7 parameters in total: SARIMA$(p, d, q)(P, D, Q)(m)$, where p, d and q are the non-seasonal parameters as above, P is the order of the seasonal AR component, D is the number of seasonal differences (by subtracting from the data the values at the seasonal-period-lag), Q is the order of the seasonal MA component, and m is the seasonal period. The new parameters are estimated similarly to the non-seasonal ones, and the modeling procedure in general is almost the same.

4.2 | Implementation

Our first implementation objective in this section is the application of the ARIMA model to the series of interest rates, potentially opting for the SARIMA extension in case a seasonality is detected in the data. No covariates are used in this part. The selection of the parameters p, d and q is performed manually, through the inspection of the series and its autocorrelation and partial autocorrelation functions, with the additional intent of better illustrating the meaning of the 3 ARIMA components.

Before doing any operation with the data, we assign the initial 80% of the series to the training set and leave the most recent 20% data points for testing. In the process of inspecting and modeling our training data, the first step is evaluating the stationarity of the series.
In Fig 4.1, we get from observing the series and its rolling mean and variance that the series is trending and not stationary. Indeed, the variance is not constant through time, and the mean fluctuates and is mostly decreasing over the considered time period; this is also backed up by the persistence of autocorrelation coefficients, in panel 4.1b. We could either be dealing with a unit root with drift or with a deterministic time trend, and the two possibilities have different implications for modeling.

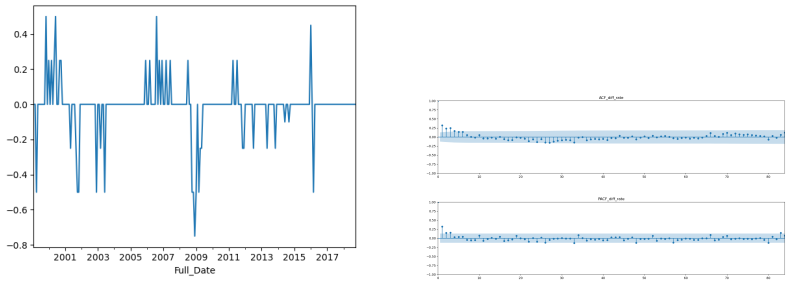
[19] A well-established statistical test called Augmented Dickey-Fuller test is useful for this purpose.



(a) Series **(b)** ACF and PACF

Figure 4.1: ECB policy rate

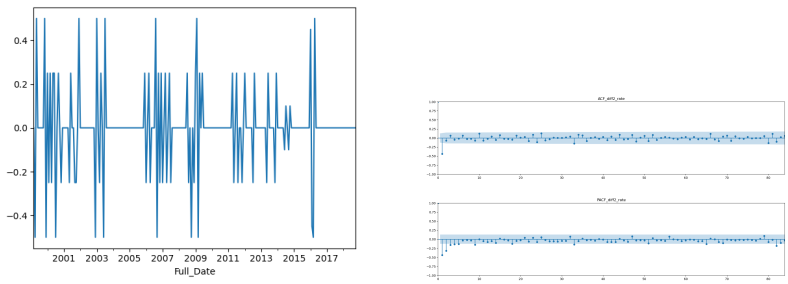
To check for the possibility of the process being a unit root process, we perform the Augmented Dickey-Fuller test. We set a threshold of 0.05 and we include a linear trend in the test regression. The p-value of 6.5% is quite small, but since it is above the threshold we set, then we cannot reject the null hypotheses and we consider the presence of a unit root. Hence, in order to make the series stationary, we try taking the first order differences. The plot of the differenced series and its ACF and PACF plots are shown in Fig 4.2.



(a) Series **(b)** ACF and PACF

Figure 4.2: Differenced ECB policy rate

The differenced series is not trending and not persistent. We test the stationarity of the series using the ADF test, and we get an extremely small p-value (around $1.9 \cdot 10^{-7}$), thereby excluding the presence of another unit root and considering the series stationary. Hence, we probably do not need to difference again and we set the ARIMA parameter d equal to 1. Now, we have to choose the order of the AR component p, and the order of the MA component q. Observing the plots in 4.2b, it is evident that there is no sign of seasonality in the series, so we do not extend the model to take a seasonal component into account. Also, we see both the ACF and the PACF having significant spikes up to lag-3, after which the ACF geometrically decays and the PACF is truncated. This type of evidence suggests 3 as the order of the AR component and 0 as the order of the MA component. Thus, $p = 3$ and $q = 0$. However, before fitting the model, we try again to difference the series, to see if we should really exclude the possibility of $d > 1$:



(a) Series **(b)** ACF and PACF

Figure 4.3: Double-differenced ECB policy rate

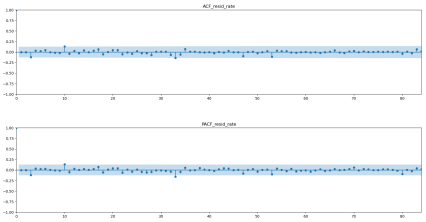
In Fig 4.3b, we see a negative lag-1 autocorrelation, indicating, as expected, that the series is probably over-differenced and so we should stick to $d = 1$. Our chosen model specification is then ARIMA$(3, 1, 0)$. At this point, we fit the model on the training data. Our estimated model is shown in Fig 4.4a.

Dep. Variable:	Interest_Rate	**No. Observations:**	238
Model:	ARIMA(3, 1, 0)	**Log Likelihood**	127.924
Date:	Mon, 05 Feb 2024	**AIC**	-247.848
Time:	12:39:52	**BIC**	-233.976
Sample:	01-01-1999	**HQIC**	-242.257
	- 10-01-2018		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.2528	0.051	4.936	0.000	0.152	0.353
ar.L2	0.1047	0.048	2.175	0.030	0.010	0.199
ar.L3	0.1595	0.049	3.226	0.001	0.063	0.256
sigma2	0.0199	0.001	19.351	0.000	0.018	0.022

Ljung-Box (L1) (Q):	0.01	**Jarque-Bera (JB):**	307.71
Prob(Q):	0.94	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.37	**Skew:**	-0.77
Prob(H) (two-sided):	0.00	**Kurtosis:**	8.36

(a) Estimated ARIMA$(3, 1, 0)$



(b) ACF and PACF of the residuals

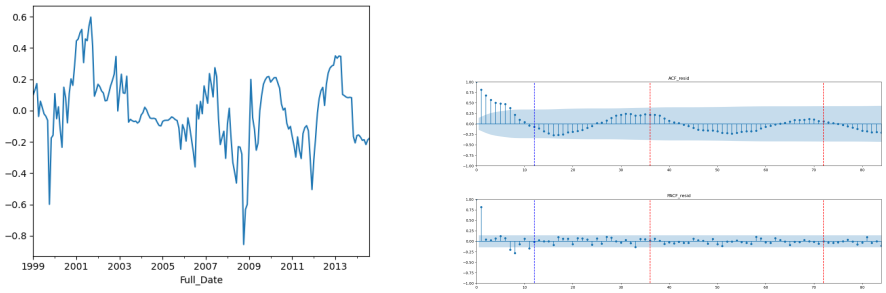
Figure 4.4: Model results

For the model to behave well, we want that the residuals of the ARIMA do not display any remaining time dependence. We carry out this check through the Ljung-Box test: since its p-value is 0.94, we cannot reject the null hypothesis of no time dependence. Thus the model residuals can be assumed to be a white noise, as desired. To the same aim, we plot the ACF and PACF of the residuals of the model in Fig 4.4b, and, as expected, they appear to be a white noise. The evaluation of the model performance on the test set is presented in the next section.

The following ARIMA implementation presents a number of differences compared to the previous one:

- We make use of the covariates which have been presented in the previous sections. We regress the interest rate on these predictors without considering the time dimension.
- The ARIMA model is implemented on the series of residuals of the above regression.
- The possible presence of a seasonal component in the series allows us to extend the model and illustrate the specification of a SARIMA. Also, multiple sets of parameters are tested.
- Since, in test-set prediction and performance evaluation, we basically assume to know the future values of the covariates, the models in this section are not intended to function on their own or provide accurate performance measures, but are meant to show the potential improvements SARIMA can stack on top of (a very performant) OLS regression, as well as to compare and illustrate different specifications.

Again, we focus just on the training data, but this time we split the series into training, validation and test sets, with 65%, 15% and 20% instances respectively. Then, we select a linear regression model by keeping only the significant variables among inflation, BCI, short-term and long-term rates. We choose to regress the policy rate on inflation and short-rates, which returns an R-squared on training data of around 0.97. We observe that, as expected, the Durbin-Watson test, which measures the degree of lag-1 autocorrelation in the residuals of the regression, has a value close to 0, indicating strong positive lag-1 autocorrelation. Our objective is to apply a (S)ARIMA model on these residuals so that all the time dependency they carry is captured and used to improve the prediction accuracy. Thus, in Fig 4.5, we plot the regression residuals, and their autocorrelation and partial autocorrelation functions, so as to choose the SARIMA parameters by inspection, as we did before:



(a) Series **(b)** ACF and PACF

Figure 4.5: Regression residuals

By the plots above, we see that the residuals show no strong trend, but the series does not seem stationary, as the autocorrelations show some persistence, and there is a clear wavelike pattern in the ACF which may indicate some seasonality component. We conduct an ADF test, which returns a p-value of around 0.2%, and so the hypothesis of a unit root is confidently rejected. Hence, we set $d = 0$: our model has no first-order differencing. Just in case, however, we try differencing the series anyway and inspect the results.

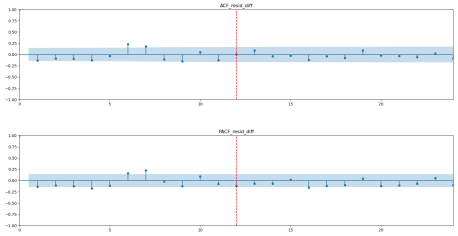


Figure 4.6: ACF and PACF of first-order differences

Confirming our decision, the plots (Fig 4.6) show a negative lag-1 autocorrelation and suggest that the series is over-differenced. Going back to inspecting the ACF of the OLS residuals (Fig 4.7), the sinusoidal pattern displayed by the coefficients may suggest that, in order to make the series stationary, we could try taking a seasonal difference.

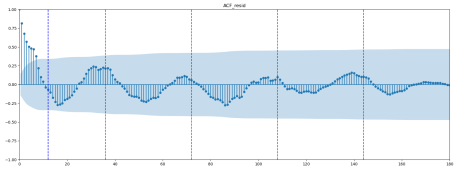
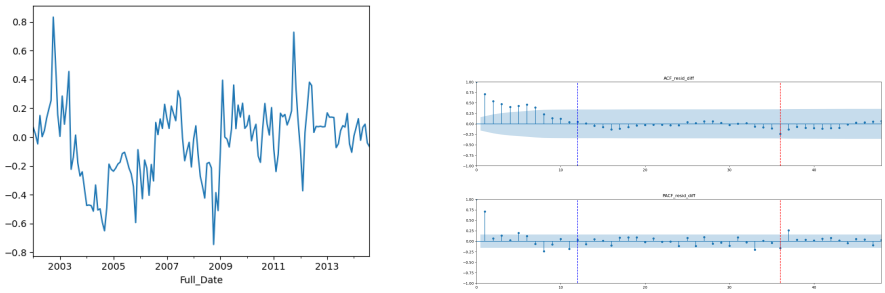


Figure 4.7: Extended ACF of OLS residuals

Indeed, although the pattern is not precisely regular, it is clear that the peaks seem to be more or less at lags of multiple of 36 months, i.e. 3 years. It is true that a seasonal period of 3 years is not usual, but this may still reflect the time interval, either fixed or average, in which the economic dynamics of the policy rate or of some covariate are fully expressed. Considering this hypothesis then, we try differencing at lag-36. Under this choice, we set the SARIMA parameters $m = 36$ and $D = 1$.



(a) Series **(b)** ACF and PACF

Figure 4.8: Seasonal-differenced residuals

Indeed, the differenced residuals in Fig 4.8 show no seasonality. Looking at Fig 4.8b, we observe the partial autocorrelation truncated at a strong lag-1, while the autocorrelation starts decaying only after lag-6. Then we choose $p = 1$ and, to make the model more parsimonious than what the plots would otherwise suggest, $q = 1$. Finally, we set $P = 0$ and, for the little spike in the ACF at the seasonal lag, $Q = 1$. Our SARIMA specification is then SARIMA$(1, 0, 1)(0, 1, 1)(36)$.

This SARIMA model is then fitted to the training data (65% of the original series), and we conduct the Ljung-Box test. This gives a p-value of 0.83, suggesting no statistically significant evidence of autocorrelation in the residuals of the SARIMA, as desired. This model however may not be the best possible SARIMA specification, and sets of parameters which differ from the one we manually specified may prove to be superior. To find alternative models, we run a grid search of SARIMAs across the following parameters space: $p \in [0, 1, 2] \times d \in [0, 1] \times q \in [0, 1, 2] \times P \in [0, 1] \times D \in [0, 1] \times Q \in [0, 1] \times m \in [36]$. For each of these 144 combinations of parameters, a SARIMA is fitted to the training set and evaluated through validation set MSE, and Akaike's Information Criterion (AIC). Finally, the two specifications which prove to be the best, one for each criterion, are saved for comparison. The main results of the grid search are the following:

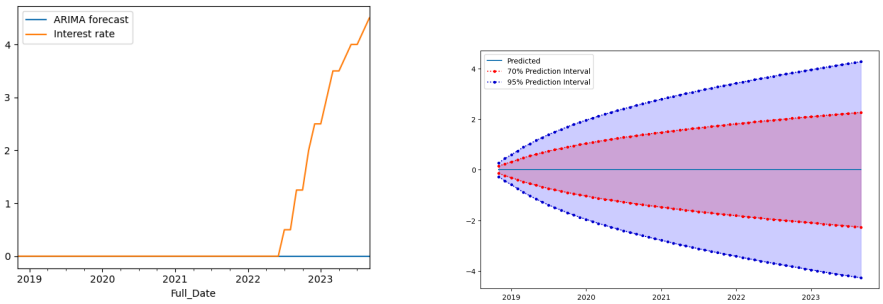
- Simple OLS, i.e. SARIMAX[20](0, 0, 0)(0, 0, 0)(36), obtained a 0.019 MSE on the validation set, better than most models.
- The best model for validation set MSE is SARIMAX(2, 0, 2)(1, 1, 1)(36), with a score of 0.014.
- The best model for AIC is SARIMAX(2, 1, 2)(0, 0, 0)(36) with a score of -233.82.
- Our manually specified model SARIMAX(1, 0, 1)(0, 1, 1)(36) has a validation set MSE of 0.029 and an AIC of -137.49.

The comparison of these models' performance on the test set is presented in the next section.

4.3 | Results

The ARIMA$(3, 1, 0)$ (without covariates) achieves a test-set MSE of around 2.34. As Fig 4.9 shows, the ARIMA model on its own is pretty limited in capturing the dynamics of the policy rate; this is likely worsened by the long, quite exceptional period that covers the end of the training data and a large section of the test data, during which the rate is fixed at 0. In Fig 4.10, we have fit the same model to the entire series, and we plotted the forecasts from Oct 2023 up to Dec 2024.

[20]With 'SARIMAX' we refer to the model obtained by stacking an OLS regression and a SARIMA model on the residuals of the regression.



(a) Predicted rate vs. True rate **(b)** Predictions with intervals

Figure 4.9: ARIMA test-set predictions

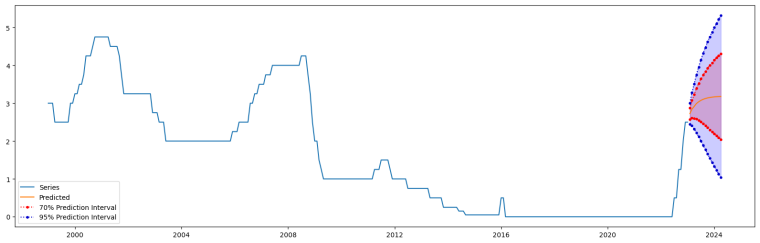


Figure 4.10: Forecasts

Finally, we deal with the performance of the models with covariates. We evaluate and compare the predictions of the 4 chosen SARIMAX models:

1. Simple OLS;
2. Manually specified SARIMAX$(1,0,1)(0,1,1)(36)$;
3. SARIMAX$(2,0,2)(1,1,1)(36)$;
4. SARIMAX$(2,1,2)(0,0,0)(36)$.

Their test-set MSE are, respectively: 0.044, 0.069, 0.093, and 0.028. From these results, we see that the two SARIMAX models that make use of the seasonal component are inferior even to simple OLS, by which it is evident that the added noise from including the seasonal orders more than compensates for whatever decrease in bias. On the contrary, SARIMAX$(2,1,2)(0,0,0)(36)$, which has been selected by lowest AIC, discarding its seasonal component achieves a better result than the other SARIMAX models and is superior to OLS. Fig 4.11 plots the predictions on the test set of all 4 models against the true rates; Fig 4.12 shows the predictions of the manually specified SARIMAX and of the best performing SARIMAX together with their prediction intervals.

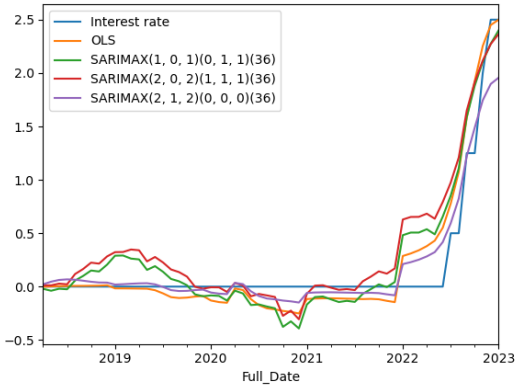
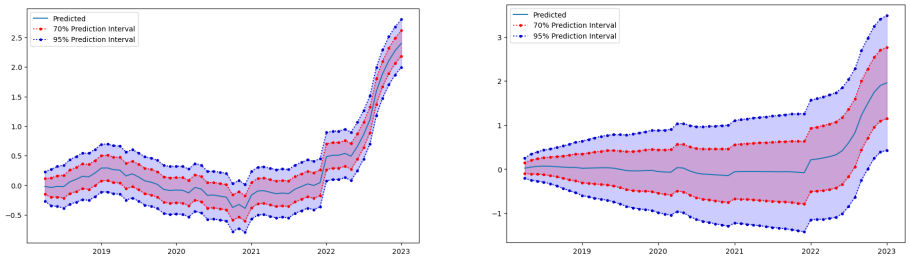


Figure 4.11: SARIMAX models test-set predictions



(a) SARIMAX$(1, 0, 1)(0, 1, 1)(36)$ **(b)** SARIMAX$(2, 1, 2)(0, 0, 0)(36)$

Figure 4.12: SARIMAX predictions

5 | Conclusions

In concluding our work, we reflect on the diverse methodologies employed in forecasting ECB Interest Rates, recognizing the intricate nature of both quantitative and qualitative economic data and their profound impact on global financial landscapes. Our journey began with the foundational approach of Linear Regression, which initially yielded promising results but prompted further exploration due to the absence of a comprehensive training-validation split.

Transitioning into the realm of statistical learning, Support Vector Regression demonstrated its efficacy in capturing and generalizing complex relationships within the dataset. The integration of K-fold cross-validation underscored the robustness of the model, as evidenced by consistently low metrics gauging the loss function.

The evolution of our analysis into time series forecasting marked a pivotal juncture, necessitating a more nuanced understanding of temporal patterns and dependencies. ARIMA models served as a fundamental tool, allowing us to unravel and model the sequential aspects of ECB Interest Rates. The subsequent incorporation of SARIMA and consideration of exogenous variables heightened the model's predictive capabilities, accounting for seasonality, trends, and external influences. As we delved deeper into the SARIMA extension, exploring various parameter combinations through a grid search, we uncovered nuances in the data's seasonal behavior and refined our model specifications accordingly. Moreover, by integrating SARIMAX models with covariates, such as inflation rates and short-term interest rates, we enriched the predictive power of our analyses, offering more comprehensive insights into the intricate dynamics of ECB Interest Rates.

Our exploration underscores the profound importance of our mission in forecasting interest rates. In a world where rates such as those set by the ECB intricately intertwine with various aspects of daily life, often shaping it, the ability to make accurate predictions holds considerable potential. The culmination of our research contributes to a growing body of knowledge aimed at informing critical decision-making processes and navigating the complexities of the global economy.

As we conclude, we acknowledge the ongoing need for adaptability and continuous improvement in modeling techniques. The dynamic nature of economic variables and their weight in the decision-making process requires a commitment to staying at the forefront of advancements in data science and machine learning. Our journey serves as a testament to the significance of rigorous analysis, collaboration, and innovation in the pursuit of accurate and impactful forecasting in the realm of interest rates. Moving forward, fostering interdisciplinary collaborations and embracing emerging technologies will be essential in navigating the ever-changing landscape of economic prediction and decision making.

6 | References

[1] ECB. Key exchange rates. *ECB website*, 2023.

[2] A. Van Der Vaart F. Bijma, M. Jonker. *An Introduction to Mathematical Statistics*. Amsterdam University Press, 2017.

[3] G. M. Jenkins G. E. P. Box. Time series analysis: Forecasting and control. *Holden-Day*, 1979.

[4] Roger Grosse. Notes on linear regression. *University of Toronto*, 2022.

[5] Xiaotong Hu. Support vector machine and its application t or machine and its application to regression and classification. *MSU Graduate Theses*, 2017.

[6] Andrew Ng. Cs229 lecture notes. *Stanford Lecture Notes*, 2019.

[7] OECD. Main economic indicators. *OECD website*, 2023.

[8] Viswa. Support vector regression: Unleashing the power of non-linear predictive modeling. *Medium*, 2023.

[9] Mingda Zhang. Lecture notes on time series: Autoregressive models. *University of Pittsburgh*, 2018.

A | Appendix A: Hyperplanes

What we considered while discussing SVR (section 3) is the notion of *hyperplane* which is deeply connected with the one of affince subspace. Consider the vector space V and $x_0 \in V$ and a subspace $U \subseteq V$. Then the subset

$$L = x_0 + U := \{x_0 + u : u \in U\} \tag{A.1}$$
$$= \{v \in V \mid \exists u \in U : v = x_0 + u\} \subseteq V \tag{A.2}$$

is called *affine subspace* or *linear manifold* of V. U is called direction or direction space, while x_0 is the support point. Note that the definition of an affine subspace excludes 0 if $x_0 \notin U$. Therefore, an affine subspace is not a (linear) subspace (vector subspace) of V for $x_0 \notin U$. Examples of affine subspaces are points, lines, and planes in $\mathbb{R}^3$, which do not (necessarily) go through the origin. Affine subspaces are often described by parameters: consider a k-dimensional affine space of the type $L = x_0 + U$ of V, then letting $(b_1, ..., b_k)$ ordered basis of U, every element $x \in L$ can be uniquely described as:

$$x = x_0 + \lambda_1 b_1 + ... + \lambda_k b_k \tag{A.3}$$

where $\lambda_1, ..., \lambda_k \in \mathbb{R}$. This is called paramatric equation of L with directional vectors $b_1, ..., b_n$ and parameters $\lambda_1, ..., \lambda_k$. With the change in dimensionality we clearly see that:

- One-dimensional affine subspaces are called lines and can be written line as $y = x_0 + \lambda b_1$. This means that a line is defined by a support point x_0 and a vector b_1 that defines the direction. See A.1

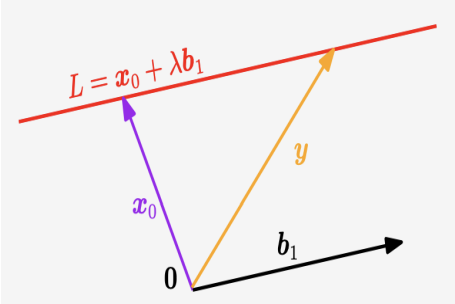


Figure A.1: Lines are affine subspaces

- Two-dimensional affine subspaces of $\mathbb{R}^n$ plane are called planes. The parametric equation for planes is $y = x_0 + \lambda_1 b_1 + \lambda_2 b_2$. This means that a plane is defined by a support point x_0 and two linearly independent vectors b_1, b_2 that span the direction space.
- In $\mathbb{R}^n$ **hyperplane** , the (n-1)-dimensional affine subspaces are called hyperplanes, and the corresponding parametric equation is:

$$y = x_0 + \sum_{i=1}^{n-1} \lambda_i b_i \tag{A.4}$$

  This means that a hyperplane is defined by a support point x_0 and (n-1) linearly independent vectors $b_1, ..., b_{n1}$ that span the direction space. In $\mathbb{R}^2$, a line is also a hyperplane. In $\mathbb{R}^3$, a plane is also a hyperplane.

B | Appendix B: Time Series Glimpse

A 'Time Series' is a collection of observations indexed by time. The observations each occur at some time t, where t belongs to the set of allowed times, T[21]:

$$\{X_t\} : \ t \in T \tag{B.1}$$

The procedure of using known data values to fit a time series with suitable model and estimating the corresponding parameters. It comprises methods that attempt to understand the nature of the time series and is often useful for future forecasting and simulation. In (B.1) we assume a time series can be defined as a collection of random variables indexed according to the order they are obtained in time, $X_1, X_2, ..., X_t$ will typically be discrete and vary over integers $t \in \mathbb{Z}$. This collection of random variables $\{X_t\}$ is referred as a stochastic process, while the observed values are as always realizations of the process. A complete description of a time series, observed as a collection of n random variables at arbitrary time points $t_1, ..., t_n$, for any positive integer n, is provided by the joint distribution function, evaluated as the probability that the values of the series are jointly less than the n constants $c_1, ..., c_n$, i.e.:

$$F_{t_1,...,t_n}(c_1, ..., c_n) = P(X_{t_1} \leq c_1, ..., X_{t_n} \leq c_n) \tag{B.2}$$

Unfortunately, these multidimensional distribution functions cannot usually be written easily. Therefore some informative descriptive measures can be useful, such as mean function[22] and more.
Lack of independence between adjacent values in time series X_s and X_t can be numerically assessed with the autocovariance function. In this sense, we define it, assuming the variance of X_t is finite as:

$$\gamma(s,t) = Cov(X_s, X_t) = \mathbb{E}[(X_s - \mu_s)(X_t - \mu_t)] \tag{B.3}$$

This autocovariance measures the linear dependence between two points on the same series observed at different times. Very smooth series exhibit autocovariance functions that stay large even when the t and s are far apart, whereas choppy series tend to have autocovariance functions that are nearly zero for large separations. Another useful and related quantity is the autocorrelation function (ACF) defined as:

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{B.4}$$

which is clearly in $[-1, 1]$ by Cauchy-Schwarz inequality. ACF measures the linear predictability of X_t using only X_s. If we can predict X_t perfectly from X_s through a linear relationship, then ACF will be either $+1$ or 1. Due to the non-determinist nature of time series, forecasting is a very complex problem. Nevertheless, we reduce its complexity by considering the stationarity of the time series, i.e. you simply predict its statistical properties will be the same in the future as they have been in the past. Mathematically speaking, a stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. Most statistical forecasting methods are, indeed, based on the assumption that the time series can be rendered *approximately stationary* after mathematical transformations. There are two types of stationarity, i.e. strictly stationary and weakly stationary:

- **strict**: if the joint distribution of $(X_{t_1}, ..., X_{t_k})$ is the same as that of $(X_{t_1+h}, ..., X_{t_k+h})$. In other words, strict stationarity means that the joint distribution only depends on the "difference" h, not the time $(t_1, ..., t_k)$
- **weak**: if
  - i. $\mathbb{E}(X_t^2) < \infty \ \forall t \in \mathbb{Z}$
  - ii. $\mathbb{E}(X_t) = \mu \ \forall t \in \mathbb{Z}$
  - iii. $\gamma_X(s,t) = \gamma_X(s+h, t+h) \ \forall s, t, h \in \mathbb{Z}$

  In other words, a weakly stationary time series X_t must have three features: finite variation, constant first moment, and that the second moment only depends on $\|t - s\|$ and not depends on s or t.

[21] Note: T can be discrete in which case we have a discrete time series, or it could be continuous in the case of continuous time series. Sometimes, we refer to one observation of the time series $\{X_t\}$ as a realisation of the series.
[22] $\mu_t = \mathbb{E}(X_t) = \int x f_t(x) dx$ provided it exists

With this notions, we apply the autocovariance and autocorrelation functions to a stationary time series (hereafter stationary is intended as weakly):

$$\gamma(t+h,t) = \gamma(h,0) = \gamma(h) \tag{B.5}$$

$$\gamma(h) = \mathbb{E}[(X_{t+h} - \mu)(X_t - \mu)] \tag{B.6}$$

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} \tag{B.7}$$

We finally define for a stationary process the partial autocorrelation function (PACF), i.e. the correlation between X_s and X_t with the linear effect of "everything in the middle" removed, as:

$$\phi_{11} = corr(X_{t+1,X-t}) = \rho_1 \tag{B.8}$$

$$\phi_{hh} = corr(X_{t+h} - \hat{X_{t+h}}, X_t - \hat{X}_t)\ h \geq 2 \tag{B.9}$$

where:

$$\hat{X_{t+h}} = \beta_1 X_{t+h-1} + \beta_2 X_{t+h-2} + ... + \beta_{h-1} X_{t+1} \tag{B.10}$$

$$\hat{X}_t = \beta_1 X_{t+1} + \beta_2 X_{t+2} + ... + \beta_{h-1} X_{t+h-1} \tag{B.11}$$