uc3m | Universidad Carlos III de Madrid

Master Degree in Statistics for Data Science
(2023-2024)

Master Thesis

“Study of air quality due to Madrid Central with Gaussian Processes”

Iván Samuel Orta Blanco

Miguel Cárdenas Montes
Juan Miguel Marín Diazaraque

Madrid, 16th September, 2024

SUMMARY

This master thesis examines the performance of Gaussian process regression models on air pollution NO_2 levels six years before and six years after the implementation of the Madrid Central. The theoretical background is reviewed, including the multivariate Gaussian distribution, stochastic Gaussian processes, and covariance functions. A methodology is presented to analyze the impact of a Low Emission Zone using meteorological data from the Retiro Park, Plaza del Carmen, and Cuatro Vientos stations. Different models are evaluated based on the predictive ability using RMSE. The methodology classifies an event as relevant if the RMSE values increase from one year to another.

Keywords: Air quality; Low Emission Zones; Gaussian Process Regression; Madrid Central; NO_2; Matern covariance functions; Rational quadratic covariance functions; relative deviations in RMSE

DEDICATION

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The implementation of Low Emission Zones (LEZ) is a matter of present day relevance. The impact of air pollutants is a critical factor for the health of the population in urban areas, and LEZs are proposed as measures to improve the air quality and mitigate the effects of air pollutants on the public health. However, the implementation of the LEZ and its effectiveness are usually questioned by the affected population and social media, since the restrictions imposed by these LEZ can pose inconveniences and challenges for the everyday routine of the people. Thus, constructing a standard methodology to evaluate the efficiency and impact of a LEZ with quantitative measures is a relevant matter.

However, the impact of the LEZ is masked by the effects of meteorological conditions from year to year. Nevertheless, segregating the influence of the weather and the LEZ in the air pollutant levels is no trivial task. In this work, Gaussian Process (GP) models that use the meteorological conditions of each day are used to model the distribution of the air pollutant NO_2. The main reference for this master thesis is the book [1], where the theoretical foundations of many GP features are described. This project takes inspiration from the methodology used in [2], where a GP model using weather variables was used to evaluate the change in nitrogen dioxide (NO_2) levels of Madrid Central (MC), the LEZ implemented in Madrid, Spain. The same case study is considered here but with a different methodology and metrics. Also different sample sizes and kernel functions were considered. Additional deviations of the GPs compared to benchmark models are computed to study how valid GPs can be as regression tools. This work also extends the years evaluated in [2], where a model trained with data from 2010 to 2017 was used to study 2018 and 2019. In contrast, in this work all the years from 2012 to 2023 are evaluated with models trained for each year (see details in Chapter 3 and Chapter 4).

The main focus of this work is to study how competent GP can be for developing a methodology applicable to any LEZ, with MC as case study. The methodology proposed in this work is only subject to the limited amount of available data regarding pollutant levels and meteorological conditions that could be retrieved for stations in to the LEZ. The NO_2 is chosen as a relevant air pollutant given that its concentration levels data was available in the stations used as data sources, but furthermore, since it is a pollutant mainly produced by car traffic. Since MC establish traffic restrictions to cars that contaminate the most, the NO_2 presents as a good option for evaluating the efficiency of MC.

The relevance of NO_2 is also shown in the World Health Organization (WHO), in the air quality guidelines from 2005 [3] and 2021 [4], were the NO_2 is presented as classic pollutant species considered in air quality studies. The impact on the NO_2 concentration on human health is also highlighted by the notable decrease on the recommended annual values of this pollutant made in the 2021 update (from $40\mu g/m^3$ to $10\mu g/m^3$).

The motivation for using GP reveals more evident in Chapter 2. The most attractive features of this statistical tools is the flexibility to model different behavior with the different choices of covariance functions. Furthermore, this tool has a clear statistical interpretation, fitting a distribution over a finite set of values of functions from stochastic processes. Many estimations for different custom loss functions could be chosen as predictions given this distribution (see Sec. 2.4 in [1]), however the mean function of this distribution is the most standard way to proceed with GP. Furthermore, confidence intervals are a natural consequence of the nature of this model. But the most distinctive feature of these models is to consider the whole dataset as just one realization from a multivariate distribution. Independence of the observations is not a mandatory assumption for GP, and the covariance between observations is modeled directly. This fact contrasts with most models in the area of supervised learning where, if statistical assumptions are made, each observation is seen as one realization of the population distribution. For instance, in linear regression the noise is assumed to be a realization of a normal distributed variable and independent for each observation. On the other hand, in contrast with models that do not assume independence between observations, such as ARIMA models for time series, the dependency between this observation is modeled directly for GP via covariance functions.

For illustrating the steps followed in this work, the structure of the document is outlined. This structure is divided into two clearly defined parts: the first part of the document is focused on describing all the relevant theory for a clear understanding of GP whereas the second part illustrates the work that has been carried out when evaluating a LEZ with the proposed methodology and results of the case study.

In Chapter 2, the mathematical foundations of GP are presented. In particular, special attention has been given to the underlying mathematics of GPs and covariance functions as their most defining element.

In Chapter 3 the steps followed to model GP for NO_2 levels is explained. In Sec. 3.1, the data retrieving process along with the problems and limitations encountered is highlighted. Next in Sec. 3.2 it is explained how the data has been preprocessed as well as the model structures considered to be competent options based on the theory explained on Chapter 2. Finally in Sec. 3.3, the considered metrics for quantifying the accuracy of the models are presented. The results and expected behavior of the metrics are presented in Chapter 4. The complete list of all codes needed to reproduce the results of this work can be accessed via the following GitHub repository:

https://github.com/ivan-samuel/Master-Thesis-2023-2024-ivan-samuel

2. THEORETICAL BACKGROUND

Although there already exist good references in the literature for understanding the underlying mathematics of GP (like [1], our main reference for this work, or more briefly in [5], [6], [7] or even [8]) this section is crucial for this work, to provide clarity for the arguments that motivate the proposed methodology as well as helping the reader to follow the conclusions and insights obtained throughout this master's thesis.

In Sec. 2.1 basics statistical tools and concepts of the machine learning are are reviewed. In Sec. 2.2 the core statistical foundations to define a GP are presented, with the additional case of GPR used for prediction, as it is the relevant application of GP for this work. Different classic assumption about the modeled noise included in observations of the GP are presented. Since covariance functions is a crucial component of GP, the Sec. 2.3 review its properties with more detail. Examples of standard valid covariance functions are presented in Sec. 2.3.1 and in Sec. 2.3.2 ways to obtain new kernels from usual covariance functions are commented. To fully specify a GP, the parameters of the covariance functions (hyperparameters) must to be determined, in Sec. 2.4 a methodological way to obtain accurate values for the data is illustrated. Finally in Sec. 2.5 an algorithm for pointwise predictions is outlined.

2.1. Review of core concepts for GPR

Before explaining GP and its application when used for regression tasks, we need to introduce some core concepts on which GP models rely. This introduction aims only to review the main and basic aspects of these concepts to understand this master thesis. Other fundamental concepts of statistics are assumed to be known by the reader, like a continuous random variable (RV) with any dimension $N > 0 \in \mathbb{N}$, the probability density function (PDF) that governs the distribution of said RV, conditional probability, marginal distribution and the mean vector of a N-dimensional RV as well as its covariance matrix (this concepts can be studied in many different references; see, e.g. [9]). The concepts reviewed here are the definition of multivariate normal distribution (MVN) and non-parametric models.

The MVN distribution could be seen as a generalization of the Gaussian or normal distribution that is widely used in many models, taken as a basic assumption of the distribution of the data as well as the noise of the data (especially when it is not clear what type of distribution our data could follow), or in many occasions, it is assumed as an approximation of the distribution.

The definition of the MVN is taken from [10] and is read as follows:

Definition 2.1.1 (Multivariate Normal Distribution). *An N-dimensional real random* *variable* $\mathbf{Y}$ *(which realizations* $\mathbf{y} \in \mathbb{R}^{\mathbf{N}}$*) with mean vector* $\boldsymbol{\mu}$ *and covariance matrix* Σ *is said to have a MVN, in symbols* $\mathbf{Y} \sim \mathcal{N}(\mu, \boldsymbol{\Sigma})$ *the PDF of* $\mathbf{Y}$ *is of the form:*

$$f(\mathbf{Y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\mathbf{1}}{(\mathbf{2}\pi)^{\mathbf{N}/\mathbf{2}}|\boldsymbol{\Sigma}|^{\mathbf{1}/\mathbf{2}}} \exp\left\{-\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{y}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-\mathbf{1}}(\mathbf{y}-\boldsymbol{\mu})\right\}$$

This MVN can be either:

i) A non-singular MVN (represented as $\mathbf{Y} \sim \mathcal{N}(\mu, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} > \mathbf{0}$*) if its covariance matrix is positive definite (p.d.), i.e.,* $\Sigma > 0$*.*

ii) A singular MVN (represented as $\mathbf{Y} \sim \mathcal{N}(\mu, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} = \mathbf{0}$*) is positive semi-definite (p.s.d.), i.e.,* $|\Sigma| = 0$*.*

Note the notation to differentiate the realizations (observed values of a RV), which are written in lowercase, from the RV itself, which is written in uppercase. The bold font represents vectors (whether they are RV or realizations) that throughout this work are understood as column vectors unless it is said the opposite.

An important property of an MVN is the marginalization property (see [1], Appendix A.2). Given a MVN random vector

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V_1} \\ \boldsymbol{V_2} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

whose realization $\boldsymbol{v} = (\boldsymbol{v_1}, \boldsymbol{v_2})$ s.t. $v \in \mathbb{R}^{d_1+d_2}, \boldsymbol{v_1} \in \mathbb{R}^{d_1}, \boldsymbol{v_2} \in \mathbb{R}^{d_2}$. Note that since Σ must be symmetric, the antidiagonal blocks are the transposes of each other, that is, $\Sigma_{12} = (\Sigma_{21})^{\mathrm{t}}$. Then it is satisfied that the marginal distribution of a subset of variables from a MVN distributed random vector is also a MVN, i.e.:

$$V_i \sim \mathcal{N}(\boldsymbol{\mu_i}, \Sigma_{ii}), \quad \text{with} \quad i = \{1, 2\}. \tag{2.1}$$

Note that if $d_i = 1$ that i-th component would be distributed as a usual Gaussian distribution with mean $\mu_i \in \mathbb{R}$ (a number instead of a d_i-vector), and variance $\Sigma_i \equiv \Sigma_{ii} \in \mathbb{R}^+$ (a number, instead of a $d_i \times d_i$ covariance matrix).

For Section 2.2 it would be convenient also to give the conditional distribution of the components of a MVN given the rest:

$$\boldsymbol{V_2}|\boldsymbol{v_1} \sim \mathcal{N}\left(\boldsymbol{\mu_2} + \Sigma_{21}\Sigma_{11}^{-1}(\boldsymbol{v_1} - \boldsymbol{\mu_1}),\, \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right). \tag{2.2}$$

Another core concept to understand GPs are non-parametric models. First, we should state that the focus of the models used in this master thesis is regression, a function that receives input from the data and tries to predict an objective continuous variable. Typically, a model is determined after incorporating information about the data that we aim to model, whether we aim to obtain valuable insights about the data or we rather aim to make predictions of the values of these data or classify data in some known categories.

In the field of machine learning, specifically in supervised learning when we have a target or response variable, Y, which is the output of the model. The inputs are realization from RV's called predictors or covariates $\boldsymbol{X} = (X_1, \ldots, X_p)$ (with $p > 0 \in \mathbb{N}$). The data that we use to determine some free parameters or hyperparameters of the model are known as training data (including both predictors and response variables). The data we use to test the performance of the model is called test data or evaluation data. An observation of the training data is a vector from the set $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$ where $\mathbf{x_i}$ is the i-th row of the matrix of observed values of predictors $X \in \mathbb{R}^{N \times p}$ (usually in $p < N$), which is known as a model matrix or a design matrix. The evaluation set is denoted in an analogous form but is labeled with an asterisk ($\boldsymbol{x_{*i}}, y_{*i}$ and $\mathcal{D}_*$).

A parametric model has a fixed number of parameters that determine the distribution assumed of the variable (for example, the population mean and variance from a Gaussian distribution). A non-parametric model could be seen as a model that has several parameters that grow in size as the training data also increase, as stated in [5]. In general, parametric models are faster and make assumptions that narrow the possibility of cases on which they should be applied, whereas non-parametric models tend to be more flexible (in the sense of having a wider range of situations where they could be competent tools) but with a higher computational demand.

However, as commented in [11], a key idea that we should take into account when opting for a parametric or non-parametric approach is if the assumptions that the method is fulfilled by the data (or at least reasonable). There exists a balance in statistics between efficiency and generality. Parametric methods favor the former sacrificing the latter, and vice versa for non-parametric methods. This is controlled by the strong assumptions about the data. In general, we could say that if the assumptions of a parametric method are fulfilled, it beats its non-parametric in terms of efficiency (this could be thought of in terms of accuracy in predictions, in the need for a smaller sample size or lower computational cost compared with non-parametric methods). However, it is important to emphasize that this is only true when the assumptions about the data are correct, and due to more restrictive assumptions, in general, this happens in a narrower spectrum of cases. The main advantage of non-parametric models is that although they do not guarantee the best results, they tend to be more accurate in a broader number of scenarios.

2.2. Gaussian Process Regression

The main reference followed, throughout this section is the classic work of Rasmussen and Williams (2006, [1]). Therefore, we present the definition of a GP provided in their book:

Definition 2.2.1 (Gaussian Process). *A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

This definition is very broad, but in the context of supervised learning, and using GP in the regression setting, we need to establish a connection between the predictors $\boldsymbol{X}$, and the response variable, Y. So a GP will be completely determined when we specify its mean function, $m(\boldsymbol{x})$, and its covariance function, $k(\boldsymbol{x}, \boldsymbol{x}')$. These functions carry out the work of introducing the information about Y throughout the knowledge of X. We will refer to a GP with the following notation:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')), \tag{2.3}$$

where $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ are defined as follows:

$$m(\boldsymbol{x}) = \mathbb{E}\left[f(\boldsymbol{x})\right], \tag{2.4}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))^{\mathrm{t}}\right], \tag{2.5}$$

Usually, the mean function is taken as $m(\boldsymbol{x}) = 0$. From now on this is assumed throughout the rest of the document, but is not needed. And in practice, is easy to implement a mean function in the GP if we have the realization of X, and we would need only to subtract this function from the realizations of $f(\boldsymbol{x})$ to have a GP that has 0 mean function ($\tilde{f}(\boldsymbol{x}) \equiv f(\boldsymbol{x}) - m(\boldsymbol{x}) \implies \tilde{m}(\boldsymbol{x}) = 0$).

In Section 2.3, we will focus on explaining the relevant information and details about covariance functions (commonly referred as kernels in the GP context). As they play a key role in the GP modeling for supervised learning. For the moment it will be convenient to introduce some notation. The covariance matrix, $K = K(X, X) \in \mathbb{R}^{N\times N}$, of a finite set of points from a GP is determined by the covariance function (2.5). Given the model matrix X then the entries of $K(X, X)$ are:

$$K_{ij} = [K(X, X)]_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_j}) \quad \text{with} \quad i, j = 1, \ldots, N. \tag{2.6}$$

To differentiate when the inputs are from the evaluation data, X_*, we use $K_{**} \equiv K(X_*, X_*) \in \mathbb{R}^{N\times N}$, $K_* \equiv K(X, X_*) \in \mathbb{R}^{N\times N_*}$ and $K_*^{\mathrm{t}} \equiv K(X_*, X) \in \mathbb{R}^{N_*\times N}$, that are defined similarly:

$$[K_{**}]_{ij} = [K(X_*, X_*)]_{ij} = k(\boldsymbol{x_{*i}}, \boldsymbol{x_{*j}}) \quad \text{with} \quad i, j = 1, \ldots, N_*, \tag{2.7}$$

$$[K_*]_{ij} = [K(X, X_*)]_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_{*j}}) \quad \text{with} \quad i = 1, \ldots, N; j = 1, \ldots, N_*, \tag{2.8}$$

$$[K_*^{\mathrm{t}}]_{ij} = [K(X_*, X)]_{ij} = k(\boldsymbol{x_{*i}}, \boldsymbol{x_j}) \quad \text{with} \quad i = 1, \ldots, N_*; j = 1, \ldots, N. \tag{2.9}$$

When using a GP to model the relation of the target variable Y with the covariates $\boldsymbol{X}$ it is typically assumed that $y = f(\boldsymbol{x})$ or that the observations are corrupted by an additive white noise (a RV distributed as a univariate Gaussian). So GP establishes a distribution over possible functions, $f(\boldsymbol{x})$, this is a case of a stochastic process. This process had labels that belong to a continuum, but when we consider only a set of finite labels (a finite number of values corresponding with its finite number of inputs) we have a MVN (see

Definition 2.1.1). The fact that we can work with a distribution that has been widely studied, and make computations of posterior, marginal, and conditional probabilities relatively easy, makes GPs a very competent tool to work with. Therefore, although a GP has an infinite number of components when we are interested in only a finite number of values of this function (which is typically the case in a supervised learning setting) we have a MVN, which is a very tractable distribution. To draw samples of these functions it is chosen a large number of input points (usually equidistant) to simulate a continuum, and then sample the values of $f(\boldsymbol{x})$ from the MVN. This technique is usually employed when making visualizations of one-dimensional GP models, to compare the functions sampled from the prior with the ones sampled from a posterior distribution in a Bayesian setting (see e.g., Fig. 2.2 from [1] or Fig. 15.2 from [5]).

A key difference from other statistical models is that usually each of the observed values of the response variable Y are seen as realizations of the same RV or from a simple random sample, srs (collection RV's independent and identically distributed, i.e., iid variables). In GP the observations are seen differently since all of the observed values are seen as only one realization from a MVN distribution, i.e., each observed value of the response variable is seen as a variable that has a covariance concerning other observations modeled by the covariance function. That is why we will represent the response variable as a vector, $\boldsymbol{Y}$, that has as many components (that are RV's) as observations.

It is evident from formulas (2.3)–(2.9) that the information that the realizations of the predictors determine the distribution of the response variable. Sometimes GP uses as predictors only one variable, e.g., time. But in general, the inputs encoded in $\boldsymbol{x}$ could be more complex, in this work they would belong to a real p-dimensional vector space (representing p predictor variables). Time would not be considered among these predictors. Since we do not make an assumption about the distribution of $\boldsymbol{X}$, and we are interested only in its observed values (just the realization and not the RV's itself), we will refer to the p model variables exclusively as covariates or predictors. This notation aims to make clear differentiation from the RV included in the MVN distribution that follows the finite observations of the GP, since each component of $\boldsymbol{Y}$ is a RV (term reserved to them).

A good property of MVN distributions is the marginalization property mentioned in (2.1). This provides GPs with a consistent study of distribution when examining more data, since the marginal distribution of a set of components of the MVN depends only on these components, and the study of more observations would maintain this distribution. This result makes consistent modeling of $f(\boldsymbol{x})$ when increasing the number of observations (i.e., the number of variables in the MVN). The examination of bigger sets does not alter how the smaller set is distributed. The consistency of the GP predictions will be discussed further in the next sections.

Prediction using training observations without noise

When we used GP to make predictions of the values of a continuous variable using the values of the predictors, we say that we are constructing a Gaussian process regressor (GPR) model. These models are considered Bayesian and non-parametric models [6]. For convenience, we will denote $\boldsymbol{f} \equiv f(\boldsymbol{x})$ as a finite number of observations of the GP.

Given a dataset of the training data $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$ and the dataset of the evaluation data $\mathcal{D}_* = \{(\boldsymbol{x_{*i}}, y_{*i})\}_{i=1}^{N_*}$ we need to establish a prior distribution. According to the predictive prior distribution (i.e., the distribution that shows our beliefs about the distribution of our data before implementing the information of the observed training data), the joint distribution of training points and evaluation points $(\boldsymbol{f}, \boldsymbol{f_*})$ is a MVN parameterized as:

$$\begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{f_*} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_* \\ K_*^{\mathrm{t}} & K_{**} \end{bmatrix}\right) \tag{2.10}$$

To make predictions we need to incorporate the information about the observed values of $\boldsymbol{f}$. In Bayesian statistics, when we update our beliefs about the distribution of our data with these observed values we obtain a so-called posterior distribution. Note that although in (2.10) we also specify the distribution that we think that $\boldsymbol{f}$ follows, the predictive prior would be understood only as the marginal of $\boldsymbol{f_*}$ (which is easily obtained looking at (2.10) and using the (2.1) property).

To obtain the posterior predictive distribution we condition $\boldsymbol{f_*}$ under the observed values of the training component. Therefore, applying (2.2):

$$\boldsymbol{f}_* \,|\, \boldsymbol{f} \sim \mathcal{N}\left(K_*^{\mathrm{t}} K^{-1} \boldsymbol{f}\,,\; K_{**} - K_*^{\mathrm{t}} K^{-1} K_*\right), \tag{2.11}$$

and to use a convenient notation, we will write the corresponding obtained posterior predictive mean and covariance matrix as:

$$\overline{\boldsymbol{f}}_* \equiv K_*^{\mathrm{t}} K^{-1} \boldsymbol{f}, \tag{2.12}$$

$$\operatorname{cov}[\boldsymbol{f}_*] \equiv \operatorname{cov}[\boldsymbol{f}_* \,|\, \boldsymbol{f}] = K_{**} - K_*^{\mathrm{t}} K^{-1} K_*. \tag{2.13}$$

In the most common cases of use of GPR, the estimator we use to predict values is the mean of (2.11). This is because we have a Gaussian distribution both mean (the value that minimizes the squared error) and median (the value of the distribution that minimizes the absolute error) coincide ([1], Sec. 2.4).

Let us inspect a little bit further into expression (2.12). We have said that our predictions for $\boldsymbol{f}_*$ in the majority of cases would be the expected value from its posterior predictive distribution $\overline{\boldsymbol{f}}_* = \mathbb{E}[\boldsymbol{f}_* \,|\, \boldsymbol{f}]$. It is interesting to inquire about how we obtain the prediction of a single observation. Having in mind this objective, let us define the vector of covariances between one observation from the test dataset concerning the rest of the training observations as a column vector $\boldsymbol{k}_* \equiv \boldsymbol{k}(X, \boldsymbol{x}_*) = (k(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}_*))^{\mathrm{t}}$. With this notation we could see the matrix K_*^{t} as a block matrix by rows, that correspond to this vector of covariances for each observation of the test data transposed:

$$K_*^{\mathrm{t}} = \begin{pmatrix} \boldsymbol{k}_{*1}^{\mathrm{t}} \\ \boldsymbol{k}_{*2}^{\mathrm{t}} \\ \vdots \\ \boldsymbol{k}_{*N_*}^{\mathrm{t}} \end{pmatrix}, \tag{2.14}$$

and so the predicted values for the i-th component of $\boldsymbol{f}_*$ could be read as:

$$\overline{f}_{*i} = \boldsymbol{k}_{*i}^{\mathrm{t}} K^{-1} \boldsymbol{f} = \sum_j^N k(\boldsymbol{x}_j, \boldsymbol{x}_{*i}) \alpha_j, \quad \text{with} \quad i = 1, \ldots, N_*; \tag{2.15}$$

where $\boldsymbol{\alpha} \equiv K^{-1}\boldsymbol{f} = (\alpha_1, \ldots, \alpha_N)^{\mathrm{t}}$. Here we recall the consistency of GPs when observing more data since we could notice that the prediction of one value, is only dependent on the training set (that defines the model completely as in any usual supervised learning model) and on the value of the test observation itself (of course the model needs to be specified, which covariance function we are assuming for our data; that would be discussed in further sections). Thus, when making predictions over bigger testing sets, once the model is fitted, said predictions would depend only on its inputs, even when we are modeling a covariance function between observations and assuming that in general, each observation is not independent of the others.

Note that including more observations in the testing, set would only affect the covariance matrix (2.13) of the MVN of the finite number of observations from the GP (since K_{**} does not appear in the predictive posterior mean). This implies that only the covariances of the newly added values of the predicted testing points are added to the covariance matrix, but not its expected values or their variances (this can be seen taking only the diagonal elements in the (2.13) formula).

Moreover, reviewing the expression (2.15) it seems clear that the smoothness of the functions that we are using to predict is conditioned by the covariance function, which, as mentioned earlier, is the key element on a GP. Once fitted, the $\boldsymbol{\alpha}$ coefficients reveal that GP models are seen as a linear combination of functions $\{k(\boldsymbol{x}_i, \cdot)\}_{i=1}^N$. That fact allows more flexibility compared with other models, e.g. linear models. For the moment, it could seem unclear which are the parameters of the model. Still, with this expression, it seems more evident that we could understand the $\boldsymbol{\alpha}$ as the model parameters, having as many parameters as observations in the training set, that is the reason in [1] (Sec. 5.4.1) the model parameters are referred as the true noiseless observations of the training set, $\boldsymbol{f}$ (often called latent variables). This interpretation is coherent with the definition of non-parametric models given before (recall [5]), as the number of model parameters increases with the number of observations in the training set.

Another important property of this approach is mentioned in the machine learning books of K. P. Murphy ([5], [6]). And that is in this approach, the GPR is also an interpolator. If we use the posterior predictive distribution mean to predict values of the training set we obtain the same observed values. That would be clear in expression (2.12) when the input is the training set, since then matrix K_*^{f} would become $K^{\mathrm{t}} = K$ and give the identity $\mathbb{I}_{N\times N}$ when multiplying with its inverse.

Prediction using training observations corrupted by noise

GPs are explained usually with a more realistic approach (see e.g. [1], [5], [6] or [7]) that is, instead of assuming that $\boldsymbol{Y} = f(\boldsymbol{x})$, to rather assume that the training observation has been corrupted by an additive white noise (a quantity that is normally distributed), $\boldsymbol{Y} = f(\boldsymbol{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n is the standard deviation of this normally distributed noise RV. This assumption gives the following joint prior distribution:

$$\begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{f_*} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} (K + \sigma_n^2 \mathbb{I}_{N\times N}) & K_* \\ K_*^{\mathrm{t}} & K_{**} \end{bmatrix}\right). \tag{2.16}$$

Applying again (2.2) we obtain the prior predictive distribution:

$$\boldsymbol{f}_* \,|\, \boldsymbol{y} \sim \mathcal{N}\left(K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}\boldsymbol{y}\;,\; K_{**} - K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}K_* \right), \tag{2.17}$$

and in this case the subsequent posterior predictive mean and covariance are:

$$\overline{\boldsymbol{f}}_* = K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}\boldsymbol{y}, \tag{2.18}$$

$$\operatorname{cov}[\boldsymbol{f}_*] = \operatorname{cov}[\boldsymbol{f}_* \,|\, \boldsymbol{y}] = K_{**} - K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}K_*. \tag{2.19}$$

Studying only one testing point gives the expression:

$$\overline{f}_{*i} = \boldsymbol{k}_{*i}^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}\boldsymbol{y} = \sum_j^N k(\boldsymbol{x}_j, \boldsymbol{x}_{*i})\alpha_j, \quad \text{with} \quad i = 1, \ldots, N_*; \tag{2.20}$$

where now the included noise its captured in the $\boldsymbol{\alpha} = (K + \sigma_n^2 \mathbb{I}_{N\times N})\boldsymbol{y}$ coefficients.

It is important to notice that the variance and covariance of the predicted variables are determined only by the inputs of the model. This is evident in the formulas (2.13) and (2.19), where only the matrices X and X_* are used and not the realization of the response variable. This is true whether we talk about noise-free realizations $\boldsymbol{f}$ or corrupted observations $\boldsymbol{y}$ (that only appear in the mean of the posterior predictive distribution). In both of the two formulas mentioned earlier, we can appreciate the contributions of two terms. The first term is just the covariance assumed by the prior distribution, and the second term subtracts from this prior belief a quantity that is inferred after fitting the model and conditioning to the observed values.

An additional clarification that needs to be made (since is the methodology that would be used in the software implementation) is about modeling the corrupted by-noise realizations of the response variable but from the test set. This approach is more appealing to the practitioner since if we are forced to work with noisy training observations, usually, the test observations are also corrupted by noise. In this case we model the joint distribution of the variables $\boldsymbol{Y} = f(\boldsymbol{x}) + \epsilon$ and $\boldsymbol{Y}_* = f(\boldsymbol{x}_*) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. The prior joint distribution is in this case:

$$\begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{Y_*} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} (K + \sigma_n^2 \mathbb{I}_{N\times N}) & K_* \\ K_*^{\mathrm{t}} & (K_{**} + \sigma_n^2 \mathbb{I}_{N_*\times N_*}) \end{bmatrix}\right). \tag{2.21}$$

The subsequent posterior mean and covariance matrix are obtained in the same way as in the former cases:

$$\overline{\boldsymbol{y}}_* \equiv \mathbb{E}[\boldsymbol{Y}_* \,|\, \boldsymbol{y}] = K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1} \boldsymbol{y}, \tag{2.22}$$

$$\operatorname{cov}[\boldsymbol{y}_*] \equiv \operatorname{cov}[\boldsymbol{Y}_* \,|\, \boldsymbol{y}] = (K_{**} + \sigma_n^2 \mathbb{I}_{N_*\times N_*}) - K_*^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1} K_*. \tag{2.23}$$

Note that when it is assumed that we have observations corrupted by noise in the training data, the GPR is not able to interpolate any longer. Thus, if we input the same values of X, different values would be predicted for the training inputs from the posterior distribution. However, if the assumptions of the model are accurate, it is expected that the GPR predicted values come close to training observed training values. These results are evident in the expressions (2.18) and (2.22), since the mixed terms from the prior covariance matrix do not include the added white noise component. The key fact that we have to take into account is that with this model, the white noise is only added regarding the labeling of the observations, and not by its values per se.

2.3. Kernels

As it seems clear now and has been said earlier, the covariance function plays a key role in determining the GP model and modeling the distribution of the data. These functions are usually called *kernels* (also covariance function, kernel function, or covariance kernel [12]). These functions are crucial not only for making predictions and modeling the distribution of the data but also for determining the relevant characteristics regarding the continuity of the sampled functions from the GP. It is commented by D. J. MacKay in [13] that if the kernel covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ is continuous on its arguments $(\boldsymbol{x}, \boldsymbol{x}')$, then the functions from the GP, $f(\boldsymbol{x})$, are also continuous. Although it is not demonstrated in its work, we could already have an intuitive idea of this by taking a look at the expressions (2.15) and (2.20). Note that the term smoothness is usually referred to as how many times a function is differentiable, not whether or not it is continuous.

There are several choices of families of covariance functions that could be chosen to formulate a GP model (in our case a GPR). In this section are presented some covariance functions that are relevant to the work presented in further chapters, It is important to remark that these covariance functions are only valid in a real vector space $\mathbb{R}^p$ for the inputs [1]. It will be seen that the conditions that need to be fulfilled to be a valid covariance function are to be symmetric on its arguments $(\boldsymbol{x}, \boldsymbol{x}')$ and positive semi-definite. Before, showing some examples of valid covariance functions, some fundamental concepts are explained.

It needs to be taken with some caution to avoid a misconception about the meaning of the covariance functions. These functions are not used to measure the similarity between two objects, they rather measure the similarity between two observed values of a function that have been evaluated in these two objects respectively [12]. The key idea behind the use of these covariance functions is precisely the concept of similarity, an intuitive idea that is supposed to be reflected in the GPR. This similarity assumption tries to achieve those observations with similar inputs have a high probability of having similar output from the model (as it would be expected in the real phenomenon that we are modeling), and therefore it is hoped that training observation would be useful to determine the value of the response variable with similar input values in the model [1].

There are three main groups of invariant covariance functions that could be mentioned: *stationary* covariance function, *dot-product* covariance functions, and *isotropic* covariance functions. A stationary kernel is a function invariant to translations to the space where the predictors $\boldsymbol{X}$ are defined, and therefore they are functions of only its difference $\boldsymbol{x} - \boldsymbol{x}'$. When the covariance function is invariant to rotations over the origin, then we treat it with a so-called dot-product covariance function, that is only dependent on the usual scalar product $\boldsymbol{x} \cdot \boldsymbol{x}'$ (although this could be generalized to scalar product defined by different inner products with different metric tensors as commented in [1]). If the covariance function is invariant to both translations and rotations (invariant to rigid motions), then it is called an isotropic covariance function, and its function only of the euclidean norm of the vector difference $r \equiv \|\boldsymbol{x} - \boldsymbol{x}'\|$.

A *kernel* is a general name, one that is used when referring to a function that maps its two input arguments into the real numbers as outcome. This nomenclature emerges in the context of the theory of integral operators [1]. Of course, this kernel function needs to fulfill some properties to be considered valid covariance functions to construct consistent GPs. The first property is the more obvious, as any covariance matrix needs to be symmetric, thus every covariance function is necessarily symmetric, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$. The next property that a kernel must fulfill to be a valid covariance function, is to be positive semi-definite, a property that arises from the Definition 2.1.1, since the covariance matrix has to be at least positive semi-definite. In the theory of integral operators, a kernel is said to be positive semi-definite if:

$$\int k(s,t) f(s) f(t) \mathrm{d}\mu(s) \mathrm{d}\mu(t) \geq 0, \tag{2.24}$$

where $s, t \in \chi$ being χ the domain of the kernel $k(s,t)$, μ a measure (see appendix A.7 from [1]) and the function $f \in L^2(\chi, \mu)$ a square integrable function. However, although the only algebraic requirement for the covariance matrix to be valid is to be positive semi-definite (a singular MVN) for defining a GP, it is not the case for constructing a GPR. As seen in expressions for the posterior predictive mean, like (2.12), to be able to make regression tasks and predictions, we need to define inverse matrices of the prior covariance matrix of the training set. This means that for GPR it is needed non-singular matrices, so the necessary condition is stronger asking for positive definiteness instead of being positive semi-definite.

Mean Square Continuity and Differentiability

This brief subsection (taken from [1], Sec. 4.1.1) aims to give a brief notion about the continuity of the stochastic processes (such as GP processes, $f(\boldsymbol{x})$). Notwithstanding that the subsequently explained mean square continuity and mean square differentiability do not guarantee the continuity or the smoothness of the sampled functions from the stochastic process.

Definition 2.3.1. *Given the sequence of elements $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots$ and one particular element $\boldsymbol{x}$ all belonging to $\mathbb{R}^p$ that satisfies the limit $|\boldsymbol{x_k} - \boldsymbol{x}| \to 0$ when $k \to \infty$ and it is also verified that:*

$$\lim_{k\to\infty} \mathbb{E}[|f(\boldsymbol{x_k}) - f(\boldsymbol{x})|^2] = 0, \quad \forall \boldsymbol{x} \in S.$$

then it is said that a function $f(\boldsymbol{x})$ is mean square differentiable in $S \subset \mathbb{R}^p$.

Note that the inputs belonging to a real p-dimensional space is an assumption that has been made throughout all the work in this document. The definition of mean square derivatives is read as follows:

Definition 2.3.2. *Given the function increment of $f(\boldsymbol{x})$ on the i-th direction, defined as $\Delta f_i(\boldsymbol{x}) := (f(\boldsymbol{x} + \boldsymbol{e}_i h) - f(\boldsymbol{x}))/h$, its mean square partial derivative of first order on the i-th is:*

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = \lim_{k\to\infty} \lim_{h\to 0} \mathbb{E}[|\Delta f_i(\boldsymbol{x_k}) - \Delta f_i(\boldsymbol{x})|^2],$$

when this limit exists.

The covariance function of the process $\partial f(\boldsymbol{x})/\partial x_i$ is $\partial^2 k(\boldsymbol{x}, \boldsymbol{x'}/\partial x_i \partial x'_i$. These derivatives can be extended for higher orders of derivation.

2.3.1. Covariance functions examples

Now we present some examples of valid covariance functions. These functions were the ones considered when carrying out the study described in the methodology (Chapter 3).

Squared Exponential

The squared exponential kernel, also called the radial-basis function (RBF) kernel, the Gaussian kernel, or the exponentiated quadratic [12], is probably the most paradigmatic kernel of all. It is usually used to introduce a GP and only one kernel function is presented. It has the following expression:

$$k(r) = \exp\left(-\frac{r^2}{2l^2}\right), \tag{2.25}$$

where l is a hyperparameter called the *characteristic length scale*, that controls how sensible is the function to the variations on its inputs difference (in one-dimensional GP, it is said that the length scale parameter controls how fast the variations on the function respect variations on the horizontal scale). It is important to note that all presented kernels in this section could be multiplied by an additional hyperparameter σ that controls the scaling of the covariance function (sometimes called vertical scaling or vertical length). The nomenclature used to refer to l and σ_s as hyperparameters is to emphasize that the model parameters are the true latent values of the GP process as discussed in the former section. The discussion about how to choose and tune these hyperparameters will be treated in Section 2.4. It is recalled the definition of $r := \|\boldsymbol{x} - \boldsymbol{x}'\|$ as the norm of vector differences that was introduced along with the isotropic covariance functions before.

This kernel is infinitely differentiable, a property that provides the GP modeled with this covariance function mean square derivatives of all orders [1]. Thus produce that the functions sampled from this GP tend to be particularly smooth. This property makes the squared exponential a very popular kernel and widely used (especially in the machine learning area).

However, this remarked smoothness is not always desirable when trying to replicate some real physical phenomena. An alternative to this election of covariance functions could be the following family of kernels, the Matern covariance functions.

There is more room for generality when defining these stationary functions choosing more hyperparameters. This could be done seen the expression (2.25) as a special case of [1]:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_s^2 \exp\left(-\frac{1}{2}(x - x')^{\mathrm{t}} M (x - x')\right), \tag{2.26}$$

where we have introduced explicitly the vertical scale σ_s and the symmetric matrix $M \in \mathbb{R}^{p\times p}$ that could have different forms depending on how we model the nature of the hyperparameters regarding each predictor and the number of hyperparameters. Some choices for this matrix are:

$$M_1 = l^{-2}\mathbb{I}_{p\times p}, \qquad M_2 = \mathrm{diag}(\boldsymbol{l}^{-2}), \qquad M_3 = \Lambda\Lambda^{\mathrm{t}} + \mathrm{diag}(\boldsymbol{l}^{-2})$$

with $\boldsymbol{l} = (l_1, \ldots, l_p)$ the possible selected length scales for each predictor, $\mathrm{diag}(\boldsymbol{l}^{-2})$ represents a diagonal matrix which entries are the values of each component of the vector $\boldsymbol{l}$ and $\Lambda \in \mathbb{R}^{p\times p'}$ with $p' < p$. However, this type of more complex specification of parameters within a covariance function family is not fully implemented in the software that we would use in Chapter 3 and 4. So for the rest of the valid kernels only the most standard forms would be presented.

Matern kernels

These kernel functions have the form:

$$k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu r}}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu r}}{l} \right), \tag{2.27}$$

where ν and l are positive hyperparameters and K_ν is a modified kernel function as stated in [1], sec 4.2.1. This family of kernels recovers the squared exponential in the limit $\nu \to \infty$.

This class of covariance functions, as long as it is true that $\nu > k$, gives to its subsequent GP the property of having mean square derivatives of order k. When the parameter ν is a half-integer number, this family of kernel functions has simpler expressions. This half-integer number could be written as $\nu = m + 1/2$, where $m \geq 0 \in \mathbb{N}$. The most interesting cases for supervised learning are mentioned by C. K. Williams and C. E. Rasmussen (in [1]), which are for the values $m = 1$ (giving GP with mean square derivatives of order one) and $m = 2$ (arising mean square derivatives of order two). Their expression is the following:

$$k_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3\nu r}}{l} \right) \exp\left(-\frac{\sqrt{3}r}{l} \right), \tag{2.28}$$

$$k_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5\nu r}}{l} + \frac{5r^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5}r}{l} \right). \tag{2.29}$$

These cases are interesting since from the collection of simpler cases with ν being half-integer, $\nu = 1/2$ gives a very rough process (note that there are no mean square derivatives that could be defined for this kernel's GP). In the case of $\nu \geq 7/2$ within the results, it is very difficult to appreciate different values between this choice of ν (especially if it is used noisy training realizations for obtaining the posterior predictive distribution).

Some families of kernels are obtained from special cases of the Matern covariance functions. The value commented before of $\nu = 1/2$ gives an exponentiated kernel $k(r) = e^{-r/l}$ that when studying one-dimensional inputs (i.e., only one predictor) is referred as the Ornstein-Uhlenbeck kernel. This kernel was used by Ornstein and Uhlenbeck to model the Brownian motion of particles as a Gaussian stochastic process (the rough sampled functions did not pose an obstacle to simulating this apparent random motion).

This former kernel, along with the widely squared exponential, could be classified in the γ-exponential family of covariance functions. The form of this class of covariance functions is:

$$k(r) = \exp\left[-\left(\frac{r}{l} \right)^{\gamma} \right], \tag{2.30}$$

where $0 < \gamma \leq 2$. However, this family is less flexible than the Matern class (2.27). This fact is mostly a consequence of the lack of GPs generated by the γ-exponential kernels that are not mean square differentiable (except when $\gamma = 2$, since we recover the squared exponential kernel).

Rational Quadratic

The rational quadratic kernel could be seen as a mixture of squared exponential kernel functions. It is a special case of a general approach to construct a valid isotropic covariance function. This procedure consists of summing infinite square exponential kernels with different values of the parameter called length scale, l, assigning different probabilities (with a PDF, $p(l)$) to each value of the length scale: $k(r) = \int p(l) \exp(-r^2/(2l^2))\mathrm{d}l$. Computing this integral when it is implemented in the PDF from a gamma distribution provides the rational quadratic kernel.

Parameterizing the squared exponential kernel with the inverse of the squared length scale $\tau = l^{-2}$ and applying the explained method provides the following result:

$$\int p(\tau \,|\, \alpha, \beta) \, \exp\left(-\frac{\tau r^2}{2}\right) \mathrm{d}\tau \;\propto \tag{2.31}$$
$$\int \tau^{\alpha-1} \exp\left(-\frac{\tau\alpha}{\beta}\right) \exp\left(-\frac{\tau r^2}{2}\right) \mathrm{d}\tau \;\propto\; \left(1 + \frac{r^2\beta}{2\alpha}\right)^{-\alpha} \,,$$

usually, the β parameter is renamed as $\beta = l^{-2}$ (this is probably due to the resemblance with the argument of the exponential in the RBF kernel).

Then the typical rational quadratic kernel has the form:

$$k(r) = \left(1 + \frac{r^2\beta}{2\alpha}\right)^{-\alpha} = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha} \,, \tag{2.32}$$

where now the hyperparameters are $\alpha, l > 0$. When $\alpha \to \infty$ the square exponential is recovered ([1], A.6). In comparison with the Matern kernel class, all rational quadratic kernels produce processes that have mean square derivatives of all orders.

White Noise

An important comment is discussed about the white noise modeled as the corrupting error from the observed values that were presented in Section 2.2. The assumption of noise in the errors is realistic in most cases (in [5], Sec. 15.2.1 are mentioned examples were assumed noiseless observation are insightful), and there could be modeled more complex structures about how this noise is modeled. Although in this work we only considered the implementation of white noise (with the Sci-Kit learn implementation), other noise models can be considered as it is commented in [13] (Sec. 5.1) or in the already very mentioned reference [1] (in the paradigmatic example of the Mauna Loa

CO_2 is modeled with the sum of two kernels, one term is the usual white noise and the other is a RBF kernel).

But even with the inclusion of a basic noise model as the white noise a GPR could become from mediocre to a very competent regression tool.

Note additionally that when making a prediction the noise is not added to the $\boldsymbol{k}_*$ vectors, since to the white noise model what matters is the **label** of the observation and not its values. Therefore we could not obtain the same values as the observed in the training observations even if the input values are the same. This encourages us to recall the commented fact that the GPR could not be used as interpolators once that white noise is added to the covariance matrix.

Periodic kernel

Another appealing kernel seems to be the one mentioned in ([13], Sec. 5.2), such is the kernel that can model functions that are periodic in intervals of λ_i, called the period of the i-th predictor. To model periodicity on each predictor $\boldsymbol{X}$, there are chosen different period coefficients, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)$. This kernel has the form:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-\frac{1}{2} \sum_i \left(\frac{\sin\left(\frac{\pi}{\lambda_i}(x_i - x_i')\right)}{l_i} \right)^2 \right], \quad \text{with} \quad i = 1, \ldots, p; \tag{2.33}$$

where l_i are different length scales chosen for each predictor as well. This kernel function should be included in the GP when working with periodic random functions, periodic components, or trends within a process.

Note that although this kernel is stationary, it is not isotropic (since it has a dependency on the vector difference but not on its norm, r).

2.3.2. Combining kernels

Another feature of GPs is that we could still obtain valid kernels by making combinations of other kernels to better adjust our model to the data distribution. This allows for extra adaptability of the GPR. This was already hinted in the brief digression about different alternatives to noise models for GP when referring to the kernel with two components from the Mauna Loa CO_2 example in [1]. And was also referenced in the immediately preceding kernel explanation, as the periodic kernel could be a valuable component of stochastic progress.

However, not every operation between kernels produces valid covariance functions. Here we present some operations [1] that can be made to obtain valid new kernels $\tilde{k}$ from already symmetric and positive semi-definite kernels, k_i with $i = 1, \ldots, n$ (being n a non-negative integer). This operations are listed below:

- Addition of kernels $\tilde{k}(\boldsymbol{x},\boldsymbol{x}') = \sum_i^n k_i(\boldsymbol{x},\boldsymbol{x}')$. Being $\tilde{k}(\boldsymbol{x},\boldsymbol{x}')$ the corresponding covariance function that define the added processes $\tilde{f}(\boldsymbol{x}) = \sum_i^n f_i(\boldsymbol{x})$.

- Product of kernels $\tilde{k}(\boldsymbol{x},\boldsymbol{x}') = \Pi_i^n k_i(\boldsymbol{x},\boldsymbol{x}')$. Note that in this case, in general, $\tilde{k}(\boldsymbol{x},\boldsymbol{x}')$ would not be the covariance function of the process $\tilde{f}(\boldsymbol{x}) = \Pi_i^n f_i(\boldsymbol{x})$.

- This operations can be extended to the direct sum $k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{x}_1,\boldsymbol{x}'_1) + k(\boldsymbol{x}_2,\boldsymbol{x}'_2)$ and tensor product $k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{x}_1,\boldsymbol{x}'_1)k(\boldsymbol{x}_2,\boldsymbol{x}'_2)$ (defined in the product space $\chi_1 \times \chi_2$), with $\boldsymbol{x} \in \chi_1 \times \chi_2$ where $\boldsymbol{x}_1 \in \chi_1$ and $\boldsymbol{x}_2 \in \chi_2$.

- With a deterministic function $u(\boldsymbol{x})$ (not a random process), we can perform a vertical scaling $g(\boldsymbol{x}) = u(\boldsymbol{x})f(\boldsymbol{x})$ from the random process $f(\boldsymbol{x})$. The corresponding covariance function would be $k(\boldsymbol{x},\boldsymbol{x}') = u(\boldsymbol{x})\,k(\boldsymbol{x},\boldsymbol{x}')\,u(\boldsymbol{x}')$.

  This allows to define normalized kernels choosing $u(\boldsymbol{x}) = k^{-1}(\boldsymbol{x},\boldsymbol{x})$, as long as $k(\boldsymbol{x},\boldsymbol{x})$ is non-negative in all its domain. This choice of scaling function would yield $\tilde{k}(\boldsymbol{x},\boldsymbol{x}') = k^{-1/2}(\boldsymbol{x},\boldsymbol{x}')\,k(\boldsymbol{x},\boldsymbol{x})\,k^{-1/2}(\boldsymbol{x}',\boldsymbol{x}')$ and when evaluating the same input it is satisfied: $\tilde{k}(\boldsymbol{x},\boldsymbol{x}) = 1$ $(\forall \boldsymbol{x})$.

- The convolution $g(\boldsymbol{x}) = \int h(\boldsymbol{x},\boldsymbol{t})f(\boldsymbol{t})\,\mathrm{d}t$ of a process $f(\boldsymbol{x})$ using a kernel $h(\boldsymbol{x},\boldsymbol{x}')$ (that does not need to be a valid covariance function) has the subsequent covariance function $\tilde{k}(\boldsymbol{x},\boldsymbol{x}') = \int h(\boldsymbol{x},\boldsymbol{t})\,k(\boldsymbol{t},\boldsymbol{t}')\,h(\boldsymbol{x}',\boldsymbol{t}')\,\mathrm{d}t\,\mathrm{d}t'$.

2.4. Hyperparameters selection

Until now we have presented the key ingredients to construct a GP and its principal features and properties. But as the reader could have noticed, to determine a GP it is needed to fix the value of the hyperparameters from each class of the covariance functions presented in Sec. 2.3 (even the omitted but mentioned vertical scale σ_s parameter that could be added to every function). Thus apart from deciding which structure we should model for the process, it is also needed to specify these hyperparameter values (that in Sec. 2.2 were assumed to be given). The task of selecting with certainty the correct hyperparameters could be possible in simpler scenarios, but in general, this is a more difficult endeavor than selecting the family of covariance functions. Sometimes the kernel class might be specified with just intuitive arguments (e.g., based on knowledge of the physical phenomenon) leading to good results in predictive performance terms (i.e., reasonably close to the expected performance of the true model covariance function). However, without a methodological procedure, the values chosen for the hyperparameters may be far from the optimal ones, even within the same kernel family. In this section, it is commented on how this decision problem is assessed.

It is important to mention that the problem of model selection is not a close topic, with room for discussion and many consolidated options. In this section, it is presented a mostly Bayesian approach with brief comments about possible alternatives.

As said before, it is crucial to specify all the details of the model to be able to make insightful usage of GPR. The model selection problem methodology implemented gives additional information about the validity of our assumptions since this model selection method could be applied in a hierarchical order to also decide from a finite set of promising covariance functions which one is the most accurate in our application.

However is important to not ignore that the final step (where the values of the hyperparameters are tuned) can also shed some light on valuable information when making inferences about the studied stochastic process. Usually, some interpretation can be obtained from the usual covariance functions. This can be seen in the recurrent hyperparameter the length scale, $\boldsymbol{l}$ (as seen in (2.26) the more general view of this parameter can be seen as a vector with one component for each predictor). This hyperparameter tells how separated two inputs have to be to make their respective observations almost uncorrelated. Bigger length scales provide less flexible sampled functions, and too short length scales tend to overfit the data with more "wiggly´´[5] functions (and with bigger posterior predictive variance). This type of analysis leads to the concept known as automatic relevance determination (ARD) which shows how inference could be made from the selection of hyperparameters. ARD has been considered in [8] for GPR, where it is attributed the origin of this ARD concept to D. J. C. MacKay and R. M. Neal. For the mentioned work is crucial the concept of likelihood to obtain the hyperparameters, is explained also in this document later in Sec. 2.4.1.

The methodology presented and followed in this work is based on Bayesian model selection, being the marginal likelihood presented below the key element for establishing the model selection procedure.

The standard way to proceed is to make hierarchical levels of inference and specify each level one at a time. The lowest level of the Bayesian model selection is the true parameters of the model, to which we will refer generally as $\boldsymbol{\theta}$. The immediate higher stage would be the selection of the hyperparameters, $\boldsymbol{\phi}$, that is involved in the distribution of the parameters, as could be seen in the (hyper) parameters of the kernels (that define the distribution of the true parameters of the model, the true latent values of $\boldsymbol{f}$). The highest level of inference determines the structure of the model, $\mathcal{H}_i$, from a finite ($i \geq 1 \in \mathbb{N}$) set of options (like choosing different kernel families). Note that the higher levels of inference need to be specified before fixing the lower levels.

For the lowest level, the posterior distribution of the parameters of the model are obtained with:

$$p(\boldsymbol{\theta} \,|\, \boldsymbol{y}, X, \boldsymbol{\phi}, \mathcal{H}_i) = \frac{p(\boldsymbol{y} \,|\, X, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathcal{H}_i)\, p(\boldsymbol{\theta} \,|\, \boldsymbol{\phi}, \mathcal{H}_i)}{p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}, \mathcal{H}_i)}, \tag{2.34}$$

where $p(\boldsymbol{y} \,|\, X, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathcal{H}_i)$ is the likelihood (the probability of obtaining the observed data assuming the model is true) and $p(\boldsymbol{\theta} \,|\, \boldsymbol{\phi}, \mathcal{H}_i)$ the prior for the parameters. This prior is chosen to be not narrow if we have little information about these parameters (like the usual expected values from an expert in the matter that could be asked). As commented in other sections, the posterior distribution updates the beliefs of the prior using the knowledge inferred from the data (through the likelihood). The denominator is a product of a normalizing constant that guarantees that (2.34) integrates to one such a proper probability distribution should satisfy. Thus it is obtained by computing the integral:

$$p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}, \mathcal{H}_i) = \int p(\boldsymbol{y} \,|\, X, \boldsymbol{\theta}, \mathcal{H}_i)\, p(\boldsymbol{\theta} \,|\, \boldsymbol{\phi}, \mathcal{H}_i)\, \mathrm{d}\boldsymbol{\theta}, \tag{2.35}$$

and this integral is called **marginal likelihood** since it is the probability of observing the given data but integrated (marginalized) over the model parameters.

The marginal likelihood plays the role of the likelihood in the posterior distribution of the hyperparameters. Thus said distribution is:

$$p(\boldsymbol{\phi} \,|\, \boldsymbol{y}, X, \mathcal{H}_i) = \frac{p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}, \mathcal{H}_i)\, p(\boldsymbol{\phi} \,|\, \mathcal{H}_i)}{p(\boldsymbol{y} \,|\, X, \mathcal{H}_i)}, \tag{2.36}$$

where now $p(\boldsymbol{\phi} \,|\, \mathcal{H}_i)$ plays the role of the prior for the hyperparameter and thus is called hyper-prior. The normalizing constant is computed in an analogous form and its expression is:

$$p(\boldsymbol{y} \,|\, X, \mathcal{H}_i) = \int p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}, \mathcal{H}_i) p(\boldsymbol{\phi} \,|\, \mathcal{H}_i)\, \mathrm{d}\boldsymbol{\phi}, \tag{2.37}$$

where again, this constant plays the role of the likelihood for the higher level of inference, the distribution of the different model structures considered:

$$p(\mathcal{H}_i \,|\, \boldsymbol{y}, X) = \frac{p(\boldsymbol{y} \,|\, X, \mathcal{H}_i)\, p(\mathcal{H}_i)}{p(\boldsymbol{y} \,|\, X)}, \tag{2.38}$$

where the normalizing constant $p(\boldsymbol{y} \,|\, X) = \sum_i [p(\boldsymbol{y} \,|\, X, \mathcal{H}_i)\, p(\mathcal{H}_i)]$ is marginalized discretely since we only consider a finite set of different model structures (the RV, $\mathcal{H}_i$, has discrete support, not a continuum). Since many of these integrals could not be solved analytically, it may be necessary to use approximation methods (such as Markov chain Monte Carlo).

2.4.1. Marginal likelihood

The marginal likelihood (2.35) would be our main tool to determine the value of the hyperparameters of the kernel functions. This methodology, taken from C. Williams and C. Rasmussen in [1], is based on the said function since they remark that the marginal likelihood has an automatic trade-off property between a good fitting of the data and model simplicity. In [1], Sec. 5.2, it is given a qualitative argument to give an intuition of this property on which this methodology relies.

This argument states that since a simple model is only useful in a limited range of scenarios and datasets, and the marginal likelihood is a PDF, then it must integrate into the unit and would be peaked over the possible dataset where it is competent. In contrast, if the model is complex (meaning, a large number of hyperparameters, usually difficult to interpret), the marginal likelihood would be more spread since it can account for more datasets and scenarios. Then, if we have a very complex model, the marginal likelihood that needs to be normalized, would have lower maximum values; on the other hand, a simpler model would have a bigger absolute maximum (this favors simplicity). However, if the model is complex, there would be more situations where the respective marginal likelihood would be higher than the values attained in the marginal likelihood of the simple model (this favors the models that can fit better the data). This trade-off will be commented on further when the explicit expression for the marginal likelihood for GPs is presented.

The Bayesian approach for model selection presented earlier is a systematic scheme to work with. Sadly, in many cases, approximations to expressions that could not be computed analytically are not trivial to derive. However, GPR constitutes an example of a method with explicit analytical expressions for the integrals described above.

As commented before, the parameters of the model seem to be hidden, and there exist different interpretations in the function space view (the methodology explained in this work) and in the weight space view (see examples in [1] or [5]). However, the former interpretation has the advantage over the latter of having always a finite number of parameters $\boldsymbol{f}$ (as long as we are working in the setting described in this work, we are usually interested in working with finite sets of data each time, recall Definition 2.2.1). The space function view could have the inconvenience of treating with an infinite number of parameters when non-degenerate valid kernels are used (see [1], sec 4.3, for details).

Let's present the marginal likelihood (2.35) applied to GPRs. The integral in this case has the form:

$$p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}) = \int p(\boldsymbol{y} \,|\, X, \boldsymbol{f})\, p(\boldsymbol{f} \,|\, \boldsymbol{\phi})\, \mathrm{d}\boldsymbol{f} \tag{2.39}$$

The prior $p(\boldsymbol{f} \,|\, \boldsymbol{\phi})$ is taken as the first diagonal block from (2.10). Note that the prior is selected only for the training data, since when want to determine the value of our hyperparameters we are "training" our model, thus only train data is expected to be used. Then the log PDF of this MVN variable:

$$\log p(\boldsymbol{f} \,|\, X, \boldsymbol{\phi}) = -\frac{1}{2}\boldsymbol{f}^{\mathrm{t}} K^{-1} \boldsymbol{f} - \frac{1}{2}\log|K| - \frac{N}{2}\log(2\pi), \tag{2.40}$$

similarly, the log marginal likelihood $p(\boldsymbol{y} \,|\, \boldsymbol{f}, \boldsymbol{\phi})$ is obtained from the first diagonal block of the prior (2.16) or (2.21):

$$\log p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{t}}(K + \sigma_n^2 \mathbb{I}_{N\times N})^{-1}\boldsymbol{y} - \frac{1}{2}\log|K + \sigma_n^2 \mathbb{I}_{N\times N}| - \frac{N}{2}\log(2\pi), \tag{2.41}$$

The marginal likelihood has three different terms, being the two first a representation of the automatic trade-off commented earlier on (the third term is just a normalizing constant). The term $-\boldsymbol{y}^{\mathrm{t}}(K + \sigma_n^2)^{-1}\boldsymbol{y}/2$ increases with the data fit [1]; while the second term $-\log|K + \sigma_n|/2$ is referred as the Occam factor by MacKay in [14] and constitutes a penalty term for the complexity of the model. Thus the mode fit–complexity model trade-off is made explicit in the marginal likelihood. Beware that the term “automatic” could be a little bit misleading, making the reader think that the optimal values for the hyperparameters for making inferences are automatically selected, this is not the case. As said before, the model selection problem is understood in a very broad sense, and thus open-ended; the marginal likelihood is a very relevant metric used in GPs for selecting these parameters, but we are not guaranteed to obtain, e.g., the best parameters for making predictions. An interesting example is shown by C. Williams and C. Rasmussen in [1] (figure 5.3 of their book) to illustrate this good behavior mentioned about the marginal likelihood.

Once the marginal likelihood is obtained, determining the hyperparameter values is somewhat direct. The procedure is to choose the hyperparameters that maximize this likelihood (a similar concept as the maximum likelihood estimator). The method is based on optimization with the marginal likelihood seen as the profit function (or the minus marginal likelihood understood as the loss function). In general, optimization algorithms benefit from explicit knowledge of the derivatives of the respective objective function. Thus the derivatives of the marginal likelihood respect the j-th hyperparameter are presented (obviously the expression is in function on the covariance matrix as each covariance function has its form). Then this general expression for the marginal likelihood is:

$$\frac{\partial \log p(\boldsymbol{y}\,|\,X, \boldsymbol{\phi})}{\partial \phi_j} = \frac{1}{2}(\boldsymbol{y}^{\mathrm{t}} K_n^{-1})\frac{\partial K_n}{\partial \phi_j}(K_n^{-1}\boldsymbol{y}) - \frac{1}{2}\mathrm{tr}\left[K_n^{-1}\frac{\partial K_n}{\partial \phi_j}\right] = \\ \frac{1}{2}\mathrm{tr}\left[\left((K_n^{-1}\boldsymbol{y})(K_n^{-1}\boldsymbol{y})^{\mathrm{t}} - K_n^{-1}\right)\frac{\partial K_n^{-1}}{\partial \phi_j}\right], \tag{2.42}$$

where K_n represents the covariance matrix with the added modeled noise (in this work it is presented only the white noise model). The number of operations involved in inverting a matrix is $O(N^3)$, being the dominant part of computing the marginal likelihood and obtaining the hyperparameters (note that this is also required for making predictions); this can be prohibitive for large datasets, being one of the biggest drawbacks of using GPs. For small and intermediate datasets GPR are more easy to implement. However, although the methodology to follow is clear, K. P. Murphy in [5] warns that the optimization problem is not convex, and thus the obtaining of a global maximum (or minimum in the case of the minus marginal likelihood) is not guaranteed for (2.41).

In this work, the same methodology is used for confronting the hyperparameter tuning, but not all the steps described in the Bayesian model selection paradigm are followed in each level of inference. In particular, the selection of the model structure differs at the top level of the inference.

The methodology followed to decide how to determine the model structure is based on predictive ability. The chosen procedure will be explained in the methodology Chapter 3. More in detail, the model structure would be decided from a set of different promising kernel families, based on the properties commented on Sec. 2.3. The paradigm chosen to decide this structure is based on estimating the generalization error [1], rather than a Bayesian approach. Note that the training error would not be used as is an optimistic measure of the model performance.

The purpose of this work for the training set and test set is not to train and test to obtain good predictive tools. The aim is to develop a methodology to evaluate LEZ and its performance (and as a consequence, it would also be measured the performance of the GPs as predictive tools). Therefore, even when the predictive ability of GPRs is also an interesting characteristic studied in this work, it is mostly to confirm whether the GPR is an adequate tool for the task or if it should be discarded in favor of more consolidated and popular tools. The aim is to infer how effective is the LEZ evaluated.

On the other hand, regarding other levels of inference, the hyperparameter tuning would be carried out by maximizing the marginal log-likelihood as explained.

2.5. Algorithm for pointwise predictions

The algorithm presented is the standard procedure for computing the posterior predictive mean (predictions), the posterior predictive covariance matrix, and the log marginal likelihood. This algorithm is from [1] (algorithm 2.1). For one input $\boldsymbol{x}_*$ and all observed training response variables realizations $\boldsymbol{y}$:

1. $L := \text{Cholesky}(K + \sigma_n^2)$ (i.e., L is lower triangular such as $K = LL^{\text{t}}$)

2. $u := \text{solve}(L, \boldsymbol{y})$ (i.e, $u \,|\, Lu = \boldsymbol{y}$)

3. $\alpha := \text{solve}(L^{\text{t}}, u)$

4. **Predictive mean value:** $\bar{y}_* = \boldsymbol{k}_*^{\text{t}} \boldsymbol{\alpha}$

5. $\boldsymbol{v} := \text{solve}(L, \boldsymbol{k}_*)$

6. **Predictive variance value:** $\text{var}(y_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) + \sigma_n^2 - \boldsymbol{v}^{\text{t}} \boldsymbol{v}$

7. **Marginal likelihood for training:**
   $\log p(\boldsymbol{y} \,|\, X, \boldsymbol{\phi}) = -0.5 \boldsymbol{y}^{\text{t}} \boldsymbol{\alpha} - \sum_i \log(L_{ii}) - 0.5 N \log(2\pi)$

Note that the heavy task that dominates this algorithm is the inversion of the matrix K, which is $O(N^3)$ of operations. This algorithm uses the Cholesky decomposition to carry out this inversion, although other methods could be chosen (like the Gaussian elimination or the LU decomposition into two triangular matrices also but one lower and the other upper). This inversion method is proposed instead of the Gaussian elimination or the usual inverse formula due to being more numerically stable and faster ([1]).

3. METHODOLOGY

As commented in Sec. 2.4, the methodology presented in this chapter is based in predictive ability for choosing the model that seems to adjust better to the data.

One of the key points of this work consists in evaluating if the GP is an adequate model to perform this task. A feature that makes these models a compelling tool is that using a Bayesian approach, we fit not only a function but a distribution that we suspect that our data could follow. Besides, we could easily obtain confidence intervals (1σ and 2σ intervals are used in Fig. 4.2) and quantiles if needed. Furthermore, the most interesting aspect of GP models is the flexibility that we have when making our assumptions about the distribution of our data by modeling different covariance matrices. These matrices as shown in (2.6) are determined by the kernel function (2.5), which is a key element for GP as explained in previous sections. The ability to model the covariance between observations contrasts with many standard models that usually assume that the observations are independent of the others. Of course, this is not the case for all the models, as it is typical in a time series setting. An example of how the assumption of dependent Gaussian observation produces smoother functions than independent Gaussian priors is shown in [7].

3.1. Data retrieving

This section aims to illustrate a transparent workflow and show problematic steps that a practitioner could encounter while reproducing the results in Chapter 4. Specific issues (as the ones regarding encoding of different data resources) can vary depending on the LEZ that the reader desire to evaluate and mostly on the country's data availability.

The data used in this work was provided from two sources: AEMET OpenData API[1] for the predictor variables and the Madrid city council [2] daily air data quality for the concentration of NO_2 levels.

The predictor variables used were taken from the AEMET API, and informative tutorials are provided to download the desired data on the mentioned web page. These tutorials are highly recommended to beginner users in the area of web scrapping. Only one issue could be non-standard to retrieve the necessary data to replicate the work presented in the following sections (or to increase the range of years studied in this work). This issue is the limitation of the range dates of the request. At the date of this work, we needed to automate two requests per year to retrieve the data of interest. This was done using Python. The details of this implementation can be checked on the GitHub repository cited in the

[1] https://opendata.aemet.es

[2] https://datos.madrid.es/portal/site/egob

introduction of this master thesis.

The response variable was not encoded in the format described in this work, where the i-th observation of the dataset $\mathcal{D}$, should be represented by a row vector. Many features, regarding the localization, and type of pollutant, were presented as rows in the Madrid city council encoding, but the time and therefore the desired value of the pollutant were encoded as columns. Additional columns were added as validation codes for missing or validated data, instead of placeholders. A more thorough study of the encoding needed for this code was needed (could be understood as a type of "transposing"). Again, details of the implementation can be checked in the GitHub repository where all codes are presented. Note that additional .csv files could be needed from the Madrid city council, from the daily air data web page.

From now on, the data is assumed to be in the format desired, that is the one presented in Sec. 2.1. The observations of the predictors are row columns in the model matrix X and the observations of the response variable are entries of a column vector. It is recalled the notation of the training and test dataset as $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$ and $\mathcal{D}_* = \{(\boldsymbol{x_{*i}}, y_{*i})\}_{i=1}^{N_*}$ respectively. Thus, retrieval of a unified dataset is used. This dataset combines both the data from the Madrid city council and AEMET OpenData API. The observations would be labeled by the day of the year. Several years would be studied, but details about training and evaluation sets would be described further in Sec. 3.2.

An important inconvenience should be commented on about this data retrieval. We would make a difference between *missing observations* and *missing values*. When referred to the i-th missing observation, it is understood as the i-th missing entry from the dataset $\mathcal{D}$, both covariates and response variable values. This type of missing data would not be imputed. However, missing values are values that are missing from one covariate within a j-th observation (a component of the j-th entry $(\boldsymbol{x}_j, y_j)$ from D). These values would be imputed column-wise, but missing data from the response variable $\boldsymbol{Y}$ provided by the Madrid city council would not be imputed, and thus the observations with missing response variable values would be treated as missing observations.

This decision was made with the understanding that the aim of the work is not only to perform well in the particular case study of MC but also to propose a standard methodology that can be applied in various situations for evaluating LEZs. Therefore, the role of the pollutant has a higher priority, and it is preferred to work with the data that is available. However, it would be explained in more detail how the covariates' missing values would be imputed since without inputs we can not ask for the outputs. The idea of using just the accessible values from the response variable is to study the real data that concerns us in LEZ, the pollutant concentrations. However, since there were no weather experts involved in this work, the methodology proposed is more flexible about the covariates available, trying to use all the information at hand and inputting missing values (or supplying missing covariates from the closest stations). Thus, the methodology is more limited to the amount of pollutant data compared to the amount of missing covariate values.

In our case study, from AEMET, there is only one observation (day) missing. However, according to Madrid City Council, there are two months of missing pollutant data in the last year evaluated, 2023. Regarding relevant missing values issues, the most problematic case was for 2021, where three wind-related covariates (direction, maximum speed, and average speed of the wind) were missing from AEMET entirely. To alleviate this problem, the decision was to supply this lack of covariate values for 2021 from the closest station. The rest of the missing values were imputed using standard methods in the area of machine learning (see Sec. 3.2). Besides these missing observations, the Madrid city council coded some values of the response variable with a validation code, `V` for valid data and `N` for negative validation. These observations were also dropped since at most there were around 20 observations in the worst cases (that is the case of 2016 and 2023).

In particular, the selected station from the Madrid city council for studying pollutant concentration was the one located in Plaza del Carmen, inside of the LEZ established by MC. However, there are no weather stations inside MC that could be of use for taking into account not only the pollutant concentration values but also the meteorological conditions (that could produce variation of said pollutant levels). Hopefully, the station of AEMET in Retiro Park is close to the limit of MC jurisdiction. This project assumes that the weather conditions under Retiro Park and Plaza del Carmen are similar enough to be informative about the behavior of the pollutant concentration. The aforementioned lack of data on relevant wind variables from AEMET is supplied with the AEMET station of Cuatro Vientos, which is the best option that could be found in terms of wind variables availability and proximity to the station AEMET Retiro station. This data supply was preferred over imputation of the commented variables since there were no values for the year 2021 and imputation seemed like an overly synthetic option.

3.2. Modeling GPRs

The models constructed of interest are as the explained GPRs of Sec. 2.2. In particular, we are interested in the case of training observation corrupted by noise, and the distribution of the response variable with the added noise. Note that the predictive mean of both $\boldsymbol{f}_*$ and $\boldsymbol{y}_*$ are the same, but the predictive variance is bigger for $\boldsymbol{y}_*$ as seen in equation (2.23) taking the diagonal (in contrast with the covariance (2.19) of the true latent variables). The response variable as mentioned before is a relevant pollutant that compromises the air quality selected from the data of Madrid city council. This pollutant is the NO_2 measured in μg/m^3 by using chemiluminescence (see the document provided by the Madrid city council: Interprete_ficheros_ calidad_del_aire_global.pdf). As stated in [4] the NO_2 is a classical pollutant in air quality studies, and although many chemical species of nitrogen oxides exist, NO_2 is the air pollutant species of most interest from the point of view of human health. Furthermore, it seems to be a suitable pollutant to measure the effectiveness of the LEZ since the principal sources of nitrogen dioxide are car traffic (and to a lesser extent industry, shipping, and households) [3] and MC aims to restrict the circulation of the most contaminant vehicles.

On the other hand, the covariates are weather variables provided by AEMET that are expected to affect the daily concentration levels of NO_2. The encoding of these AEMET variables is presented from the metadata provided by the API:

- `tmed`: Average daily temperature. Units: Celsius degrees (ºC).
- `prec`: Daily precipitation, from "07 to 07" (understood as from 07:00 AM to 07:00 AM). Unit: millimeters (mm). Special placeholder `'Ip'` is encoded for values lower than 0,1 mm. There is mentioned another special value `Acum` meaning accumulated precipitation, but it was not observed in any of the data requested. Note that this is a volume measure used in pluviometry and not a length unit since it refers to the height of a quadrangular prism of a base area of 1 m^2.
- `tmin`: Minimum temperature of the day. Unit: Celsius degrees (ºC).
- `tmax`: Maximum temperature of the day. Unit: Celsius degrees (ºC).
- `dir`: Direction of the maximum wind gust. Units: tens of degrees. It is used the following placeholders for 99 = *variable direction* and 88 = *no data*. Note these values, 990º and 880º, are not defined if we restrict to the interval 0º to 360º.
- `velmedia`: Average wind speed. Units: meters per second (m/s).
- `racha`: Maximum wind speed. Units: meters per second (m/s).
- `presMax`: Maximum pressure at the reference level of the station. Units: hectopascals (hPa).
- `presMin`: Minimum pressure at the reference level of the station. Units: hectopascals (hPa).
- `hrMedia`: Average relative daily humidity. Measured in percentages (%).
- `hrMax`: Maximum relative daily humidity. Measured in percentages (%).
- `hrMin`: Minimum relative daily humidity. Measured in percentages (%).

Note that time is not a variable of the model and is used just as a label to define an intuitive and natural order for the observations. Therefore, the Fig. 4.2 is just a visualization of the performance of the models and does not represent the sampled function from the stochastic GP.

Now is the time to discuss how the models are implemented and evaluated. One of the options considered to evaluate the performance of the models was to train with data from years before the implementation of MC (2018), and then evaluate if the ongoing years seemed to adapt to the fitted distribution previous to MC. Then it could be inferred if the effect of MC was relevant or not. The problem spotted with this approach was that car models would not be considered in any way. Thus this approach could be argued that spot differences between training observations and evaluation data in a more biased way since new car models are expected to contaminate less. However this fact has not been demonstrated to be a determinant factor in the performance of the GPR, it is just a motivation that leads the author to discard this option.

The approach followed was to study each pair of years consecutively. The earlier year is used for training and the following year is used as the evaluation set (as demonstrated in Fig. 4.2). It is expected that following this way of procedure, the changes in the car models from one year to the next are not enough to make the predictions always inconsistent. Thus it is assumed that if no major events are altering the distribution of the pollutants the GPR trained will have a good performance predicting the subsequent year. This approach seemed more fair than the one commented before and thus is preferred to avoid a natural bias in the procedure of evaluating a LEZ.

The commented partition of train and test data has an additional advantage, and that is that the volume of data is tractable by most of the models and algorithms that one could think of, since as maximum there would be only 366 observations in the train year if it is a leap year. GPRs greatly benefit in training with a relatively small amount of observations, since it scales as $O(N^3)$ due to the inversion of covariance matrices.

The steps followed for constructing the models are the standard for supervised learning: imputation of missing data, preprocessing, training, and evaluation of the model.

3.2.1. Implementation

The software used for this task was Python 3.10.0, it is worth commenting that the codes have been run on other machines with Python 3.12.6 and no major issues have been detected. However (especially for the .ipynb for training models in the GitHub repository) there has not been an intense bug search for this version, and the recommended version for reproducing the results is Python 3.10.0.

The main library that carries out the framework described in Chapter 2 is Scikit-learn 1.5.1 [3]. Other core libraries are used such as NumPy 1.24.3 or Pandas 2.2.2.

Imputation

The imputation step was carried out on two levels: one for the special encoding that is provided by AEMET metadata and presented in Sec. 3.2; and one for data that is not available without knowing the reason (these type of not available data will be referred as NA).

[3] https://scikit-learn.org/stable/modules/gaussian_process.html

In the case of the covariate `prec`, the placeholder `Ip` has a clear meaning. These values are precipitations below 0.1 mm. So it seems reasonable to impute this value by just selecting one value from the interval $[0, 0.1)$. However we have no additional information to select that value, and it is hoped that any value inside of the commented interval provides similar performance of the model. It seems, studying the rest of the values of the variable `prec` that 0.1 mm is the precision of the device or tools used to measure the precipitation levels. Therefore a placeholder is used to inform values greater than 0 and lower than 0.1 mm. Making this assumption, it was decided to take any value below the precision of the device as 0.

The other predictor that we should take care of how it is imputed is `dir`. In this case, we have two different encoded values with different meanings. The value 88 would be the easier one to treat since it represents a usual NA in the context of machine learning and it would be imputed the same as the rest of the missing values in the second level of imputation. However, the values encoded as 99 represent variable directions in which there has been registered the maximum speed of the wind. Therefore, the imputation is selected differently. Even if it could be imputed as the rest of NA the following procedure seemed more accurate knowing the nature of the missing data. This procedure assumes that if there were many directions in which the maximum wind speed was registered, then it was probably a mixture of the most frequent values for the direction of maximum speed. Thus, an average of the `dir` column for each year is imputed in each of 99 values encoded. This procedure hopes to capture the behavior of a mixture (understood as an average) of different wind directions. More sophisticated imputation methods could be studied, but the dependency of the imputation method on the results was not studied in this work.

It is also worth mentioning that there were predictors provided by the AEMET Open-Data API that were not used. These variables represent the time hours of the day in which the maximum or minimum value of some of the mentioned covariates is reached. The encoding would need to be a little more detailed, a possible option was to convert the time hours into a continuum variable, changing the units into seconds, inside the interval $[0, 86400]$s (the seconds within a day). However, since there was a considerable proportion of missing data within these variables, it would need a great amount of imputed values in each year. In the less severe cases (a few of the last years) this imputation would be carried out in approximately a ∼35% of the observations, but in most cases, the imputation represents percentages between 60% and 70%. Thus it was decided to discard these variables with many NAs.

The second level of imputation was carried out precisely on NA values, where no additional information was provided about the nature of the missing data. This imputation was performed with the K-Nearest-Neighbors (KNN) algorithm with K=10. The selected value of K was chosen in a rather arbitrary way, other values such as K=5 (default value of Scikit-learn) could be chosen. Since there is no optimal way to select this value usually a grid search (e.g. via `sklearn.model_selection.GridSearchCV`) is performed to optimize the value of imputation. However, it seemed difficult to obtain some values of the trained model in Scikit-learn when tuning the hyperparameters of the imputation tool, since these would require the use of a `sklearn.pipeline.Pipeline` class that has different attributes than the class `sklearn.gaussian_process.GaussianProcessRegressor` which implements GPR. Note that implementing this exhaustive imputation grid search would also imply choosing the optimal value K for imputing each of the models commented in Sec. 3.3.1 to make a fair comparison. In this work only one value of K is selected for the imputation of missing values using KNN and all models would receive the same imputed and preprocessed data.

Recall that in the case of NA values for the response pollutant variable, no imputation was carried out. It was decided to work with the data available for the response variable and not to ask for predicting these values (since we have no reference for a true value) nor to train any model making use of these observations.

Preprocessing

Before training the models we have to be sure that the data is suited for the task. In Fig. 3.1 (a) is shown that the data is not commensurate, i.e., the different predictors do not have comparable scale in range of values or in variance. The predictors `presMax` and `presMin` have greatly larger values compared with other magnitudes included in the model. To avoid that the model being dominated by only one or two of the 12 features preprocessing seems to be mandatory.

One option could be only to center the values of `preMax` and `presMin` around a closer value of the mean of the rest of the predictors. However, taking a look at Fig. 3.1 (b) we can see that even without the pressure-related magnitudes the predictors are not commensurate. Thus a scaling applied to all covariates is preferred.

One needs to be careful when scaling the data since data leakage must be avoided. Data leakage occurs when information from the evaluation set is filtered into the training process. Only information on the training set should be used in the training of the models to have honest results and realistic performances of the models. Thus when training only the train data must be used and no information from the evaluation set must be seen by the model.

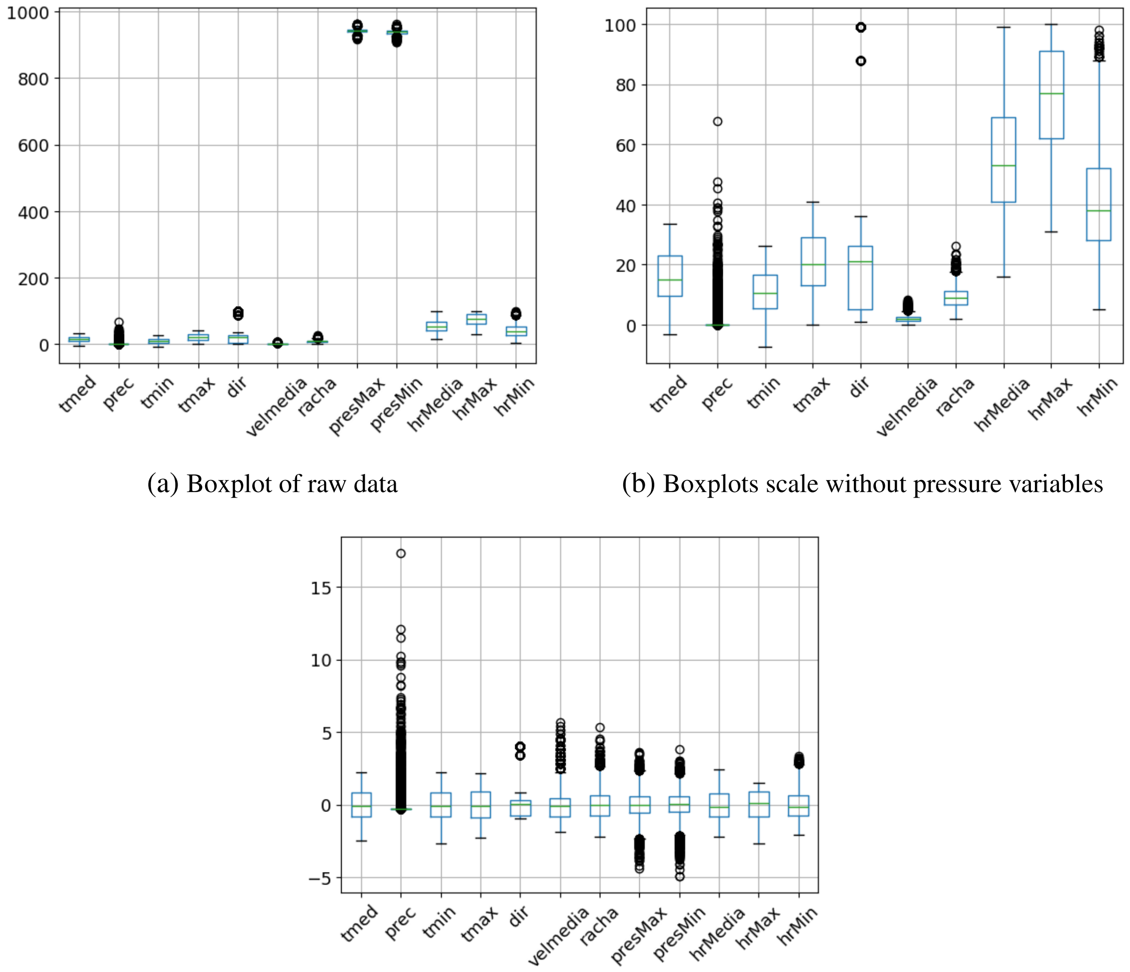
The scaling used was the standard scaling, performed by the operation:

$$Z_i = \frac{X_i - \mu_{xi}}{\sigma_{xi}}, \tag{3.1}$$

where the X_i in this context represent one predictor magnitude, μ_{xi} represent the sample mean measured from the observations and σ_{xi} the sampled standard deviation. The operation must be applied to all the observed values within the train data. Standardizing the data provides a sample with zero mean and unit variance.

With this operation we have successfully obtained commensurate data, as seen in Fig.

3.1 (c). Nevertheless, we can observe that the variable `prec` predictor has many outliers. This fact is due to the massive amount of days with no rain in Madrid, so it is normal that this variable is very skewed.



(a) Boxplot of raw data

(b) Boxplots scale without pressure variables

(c) Boxplots after scaling the data

Fig. 3.1. Boxplots showing the scale on the variability of each covariate. The first boxplot shows all the data without any preprocessing, it is clear that the pressure-related predictors have a very different scale compared with the other covariates. In the second plot is appreciated that even without showing the pressure covariates the rest of the predictors seem to have very different scales. The last plot shows the data but scaled. Note that these boxplots do not represent the inputs of the model and are just a visualization of all the data from all the years requested from the AEMET OpenData API. Each model would scale its inputs regarding only its corresponding training data.

Discussion of considered kernels

Given the properties commented in [1] that are presented in the sec. 2.3 of this work, two kernels arise as noteworthy options. These kernels are the Matern family class of covariance functions and the rational quadratic kernel.

From the Matern family class the most interesting options are the ones recommended in [1], for $\nu = 3/2$ and $\nu = 5/2$ being mean square differentiable one time for the former and two times for the latter. This class of covariance functions is appealing in real data case studies as they are presented as an alternative to the RBF kernel. This is mostly because RBF kernel produce very smooth processes that could not adjust to real physical phenomenons (that are usually also corrupted by noise). However, the Matern covariance does not produce stochastic processes that are infinitely mean square differentiable (recall Definition 2.3.2), but still has some mean square derivatives for its stochastic processes. As mentioned before, exactly the mean square derivatives of order one for $\nu = 3/2$ and order two mean square derivatives for $\nu = 5/2$.

The rational quadratic kernel results are interesting as they arise from a mixture of infinite RBF kernels with different length scales, where each length scale has a weight given by a gamma distribution. Although no demonstration is provided in [1], intuitively we could think that this kernel is a good alternative to RBF hoping that the mixture of different RBF kernels helps to make the processes more flexible to the data somehow related to the many possible length scales considered while integrating in 2.3.1. Furthermore, the stochastic processes that arise from the rational quadratic kernel are also infinitely mean square differentiable in contrast to the Matern class of kernels that always have a finite number of existing mean square derivatives for their corresponding processes.

Although these two families of covariance functions are the ones suspected to be the best for suiting our case study, other classes of kernel functions have been considered. All the considered options were implemented in Scikit-learn within the parent class: `sklearn.gaussian_process.kernels`.

These kernels were the exponential squared kernel (`RBF` in Scikit-learn), the Matern $\nu = 3/2$ (`Matern(nu=1.5)`), the Matern $\nu = 5/2$ (`Matern(nu=2.5)`), the rational quadratic kernel (`RationalQuadratic`), the `ExpSineSquared` and the sum of a RBF kernel plus `ExpSineSquared` (employing the properties explained in Sec. 2.3.2).

Note that the `ExpSineSquared` (called from now on *sk-periodic* kernel) is not introduced in Sec. 2.3. It could seem similar to the presented periodic kernel (2.33) taken from [13] but as reviewed in the current documentation of Scikit-learn it is not the same expression. The sk-periodic implementation reads as:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-2\frac{\sin^2(\pi\|\boldsymbol{x} - \boldsymbol{x}'\|/\lambda)}{l^2}\right), \tag{3.2}$$

where this implementation only consider the same parameter λ and l for each predictor. As it is noticed, both kernels are stationary but only the sk-periodic kernel is isotropic (recall Sec. 2.3). For this implementation to be equivalent, instead of the norm of the vector difference, the sine of each component should be added inside of the exponential. No further references were found in the Scikit-learn documentation for the study the properties of this kernel.

All the kernels tested in this work had been considered with the additional hyperparameter commented in Sec. 2.3.1, the vertical scale. And when combining the RBF kernel and the sk-periodic kernel each component was provided with its vertical scale. Furthermore, in all tested kernels the white noise assumption was made since it was mandatory to obtain reasonable predictions using a GPR for this case study.

That said, it should be mentioned that although all these models can be seen in the best kernel candidates search in the GitHub repository. In Chapter 4 only the models with the best performance are shown to avoid overinterpretation and to be concise and show clear and transparent results. Therefore only the Matern $\nu = 3/2$ and the rational quadratic kernel are presented. Since they were the best two performers among the rest of the kernels considered.

Now is pertinent to comment on the introduction of noise in the model. The implementation of Scikit-learn allows to implementation of white noise in the observations with an object of the same class parent class as the kernels mentioned before, the `WhiteKernel`. However, to obtain the posterior predictive covariance matrix of training observations corrupted by noise but not the test data, as seen in the expression (2.19), one should need to subtract the noise component manually. By default Scikit-learn provides the formula (2.23), as can be corroborated looking at the source code[4] of the `WhiteKernel` and the attribute `.predict()` from a GPR in Scikit-learn.

There is no need to do a standard grid search implementation for setting the correct hyperparameters of the kernel function. This task is already carried out by Scikit-learn, implementing the log marginal likelihood described in Sec. 2.4.1 as an objective function. Optimization algorithms implemented in the library look and search for the hyperparameter values that maximize this function. The user is allowed to play with the number of times that the optimization problem is restarted on different points. Besides the software gives the option to pass a custom optimizer following the syntax explained in Scikit-learn documentation. Bounds for the possible values of hyperparameters can also be chosen before the training process begins.

3.3. Metrics used to measure the performance of the model

In Sec. 2.4.1 it was already commented that inference about the highest level of model selection would be carried out regarding the predictive ability of the models. This type of model selection is called in [1] as estimating the generalization error. This type of model selection problem usually are more robust than the Bayesian approach described in Sec. 2.4 for selecting the model structure when there is model misspecification.

So different functions would be used for measuring the performance of the model when predicting the values of the test response variable. This type of function is called *metrics* in the area of machine learning. The metrics considered in this work are the

[4] See source code for the noise model component of the kernel here and here for the GPR class.

likelihood of a MVN, the logarithm of said likelihood, and the usual root mean squared error (RMSE).

The log-likelihood of a general MVN is:

$$\log \mathcal{L} = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) - \frac{N}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| \tag{3.3}$$

This expression can be read from Definition 2.1.1 just by taking the logarithm of the PDF of the MVN distribution. Note that the mean vector μ that we have to input in the expression (3.3) is the of the predictive posterior mean vector of the distribution of the test values $\boldsymbol{y}_*$, i.e., the vector defined in the expression (2.22). Thus the covariance matrix Σ in the case of GPRs must be the the covariance matrix introduced in (2.23). Naturally, in our case of interest, the $\boldsymbol{y}$ represents the observed values of $\boldsymbol{y}_*$ and the dimension N of the MVN is N_*.

Note that this function should not be mistaken by any means for the log marginal likelihood presented in Sec. 2.4.1. Although both expressions represent the log-likelihood function of a MVN, in Sec. 2.4.1 the inputs are related to the training set (since this is the objective function optimized in the training problem). In contrast, when we refer to just the “log-likelihood”, $\log \mathcal{L}$, we are implying that the model performance is being tested using the testing set as inputs (since the evaluation problem is the one being assessed this time).

The broadly used and well-known RMSE function reads as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N_*}\sum_{i=1}^{N_*}\left(y_i^{\text{true}} - y_i^{\text{pred}}\right)}. \tag{3.4}$$

In contrast with the log-likelihood presented in the equation (3.3), the RMSE is seen as a loss function. Therefore the scope when using as a metric the $\log \mathcal{L}$ is to maximize whereas with the RMSE is to minimize. Better models should be the ones that obtain higher values for the log-likelihood and lower values of the RMSE.

It is worth commenting that the likelihood is a more limited metric compared with the log-likelihood in computational terms. Since the parameter of the dimension of the MVN is involved in (3.3), when taking the exponential the limited memory of the software tends to store the value of the likelihood as the extremes of the exponential: 0 for small values of the $\log \mathcal{L}$ and the infinite code of the software (`numpy.inf` in the NumPy library).

Note that the expression (3.3) is the logarithm of the probability of observing the data given that the model is true. Thus this metric is only valid for models that assume a MVN distribution (such us GPs). The RMSE on the other hand tries to estimate the (squared) error that the model makes when predicting the test data, and only requires of the predicted values of the model. Thus the RMSE is a universal metric for every model used in a regression setting.

3.3.1. Improvement compared with simpler models

In addition to the discussion made for measuring the performance of the GPR models, a comparison is made with the RMSE obtained with other widely known models.

These models were chosen to be linear since the aim of said models is to serve as a benchmark. This comparison would provide us with useful information about the advantages and problems of using GP models, compared with well-established simpler models. The intention is to measure the predictive ability of GPRs and evaluate if the added complexity is worth it both in the needed theoretical background and computational time.

The first benchmark model chosen is the ordinary least squares (OLS) in the frequentist approach. This model is well known as the pioneer in the field of the regression problem. The main drawback of OLS is the assumption that the predictors and the response variable share a linear relationship, which in general is considered as a lack of flexibility for this type of model. The objective when fitting a OLS regressor is to obtain the hyper-plane that minimizes the RMSE with the observed training data. The assumed relation between the response variables are the predictors is:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + +\beta_{p-1} X_{p-1} + \beta_p X_p + \epsilon, \tag{3.5}$$

where p (following the notation of this work) represents the number of predictors and ϵ is a white noise error component that is normally distributed with zero mean. The $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ parameters are the $(p+1)$ parameters of the model, these constants are selected to obtain the lowest training RMSE when the assumptions of the model are satisfied (which can be reviewed in [5]). The estimates of these coefficients are computed as follows:

$$\hat{\boldsymbol{\beta}} = (\tilde{X}^{\mathrm{t}} \tilde{X})^{-1} \tilde{X}^{\mathrm{t}} \boldsymbol{y} \tag{3.6}$$

where the matrix $\tilde{X}$ is the matrix X but with an added column $\mathbf{1} = (1, \ldots, 1)^{\mathrm{t}} \in \mathbb{R}^{p+1}$ of ones:

$$\tilde{X} = (\mathbf{1} \,|\, X)\,.$$

The predicted values for one point of the test data are:

$$\hat{y}_* = \beta_0 + \sum_{i=1}^{p} \beta_i x_*. \tag{3.7}$$

Many tools have been proposed to beat OLS. In particular, shrinkage methods present themselves as natural alternatives to OLS when predictors exhibit high correlations and when the number of predictors is higher than the number of observations in the training set. These methods are usually understood as linear regression such as OLS, but with an added penalty term to the loss function. Two of the most popular shrinkage methods are ridge regression and the LASSO (see a comparison between both methods in the classic paper [15] where LASSO is introduced). In this work, it is used a combination of both of these penalties with the elastic net model (see [16]). These methods do not have an explicit solution for the coefficients of the linear models, but by using optimization algorithms for the respective loss functions of each model solutions can be obtained.

The OLS and the elastic net models would be fitted for every year, and their performance with the RMSE would be compared to the different modeled GPRs in terms of RMSE. Since OLS is usually understood as a basic tool, the behavior expected for good models is that they beat in predictive ability the OLS models. The elastic net is a more sophisticated tool and is usually more flexible in a broader range of scenarios, so a good model would be expected to be close to the performance of the elastic net.

4. RESULTS

This chapter presents the results obtained from applying the proposed methodology in Chapter 3. In this section, the results are presented and described, but the conclusions would be left for Chapter 5.

First of all, the most basic information that could be obtained is to observe the mean of the concentration values of NO_2 in each year. In this work, 12 years have been studied to infer if MC had a relevant effect on the distribution of the chosen pollutant. Since data from six years affected by MC was available (from 2018–2023), it was chosen the same number of previous years without being affected by its implementation (from 2012 to 2017). However, 2018 was only affected starting from November.

Note that the year used for training is the one before the year being evaluated. Thus the training period covers the years 2011–2022 and the years evaluated are 2012–2023.

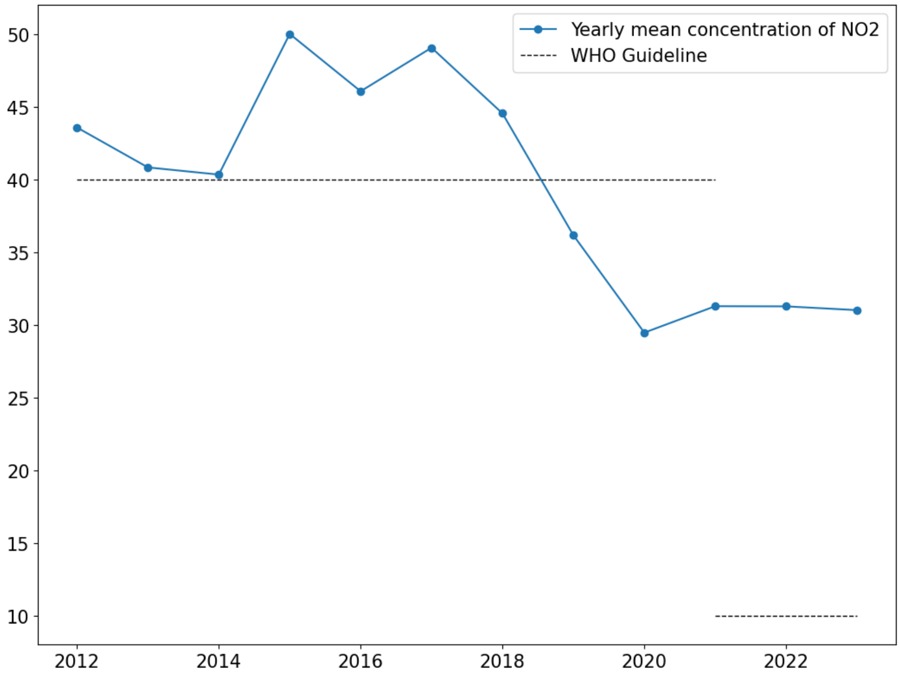
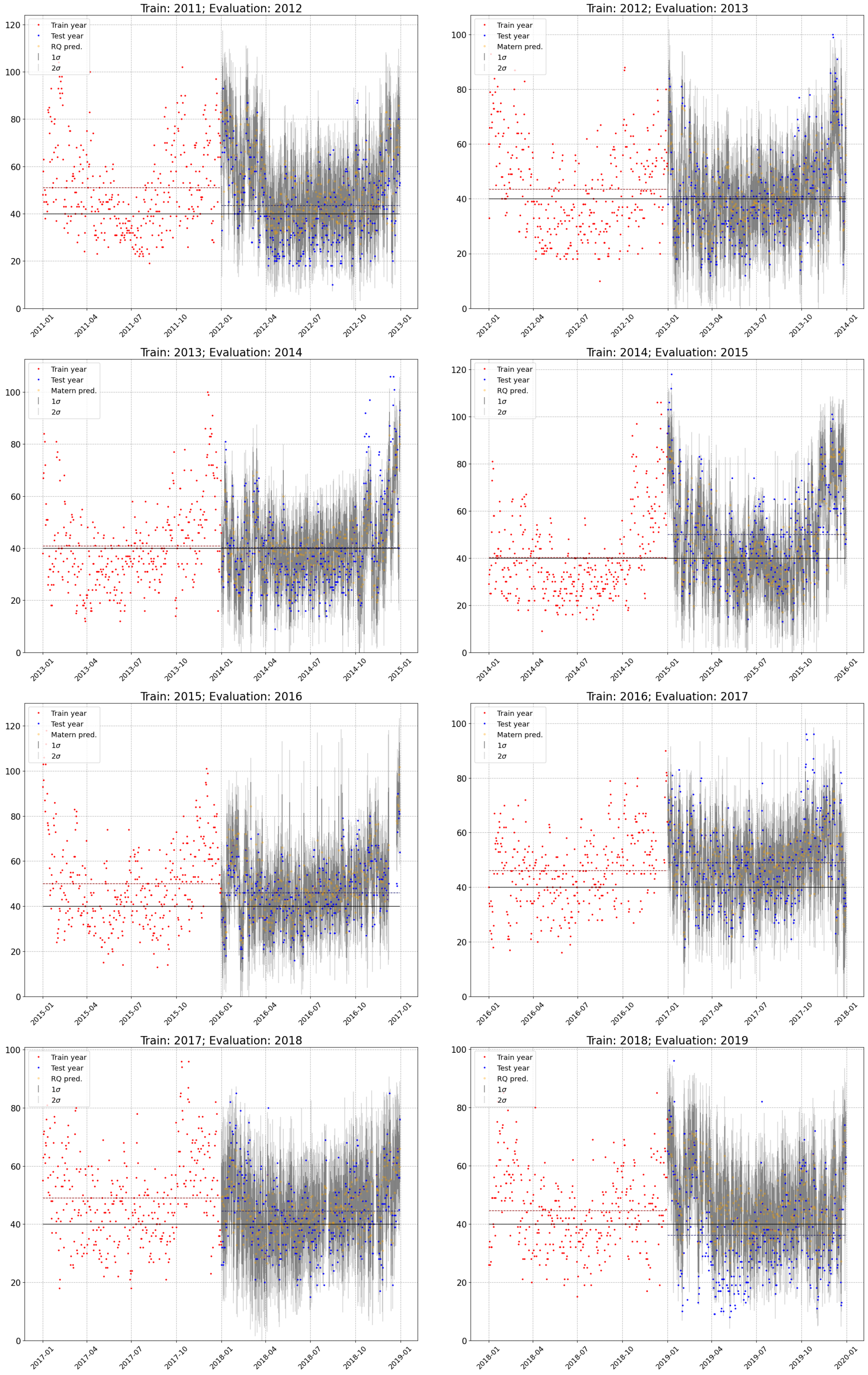


Fig. 4.1. Observed annual mean of the NO_2 concentration throughout the years. Horizontal dotted lines are also drawn to illustrate the recommended yearly average values of NO_2 by the WHO guidelines. This values are taken from the 2005 [3] and 2021 [4] updates

Taking a look at Figure 4.1, it seems that the increasing tendency from 2015 started to lower its levels after 2018. This decrease was observed until 2020 when the COVID-19 pandemic took place. Afterward, the concentration of NO_2 levels appears to have stabilized until 2023 without drastic increases or decreases.


Train: 2011; Evaluation: 2012

Train: 2012; Evaluation: 2013

Train: 2013; Evaluation: 2014

Train: 2014; Evaluation: 2015

Train: 2015; Evaluation: 2016

Train: 2016; Evaluation: 2017

Train: 2017; Evaluation: 2018

Train: 2018; Evaluation: 2019

Fig. 4.2

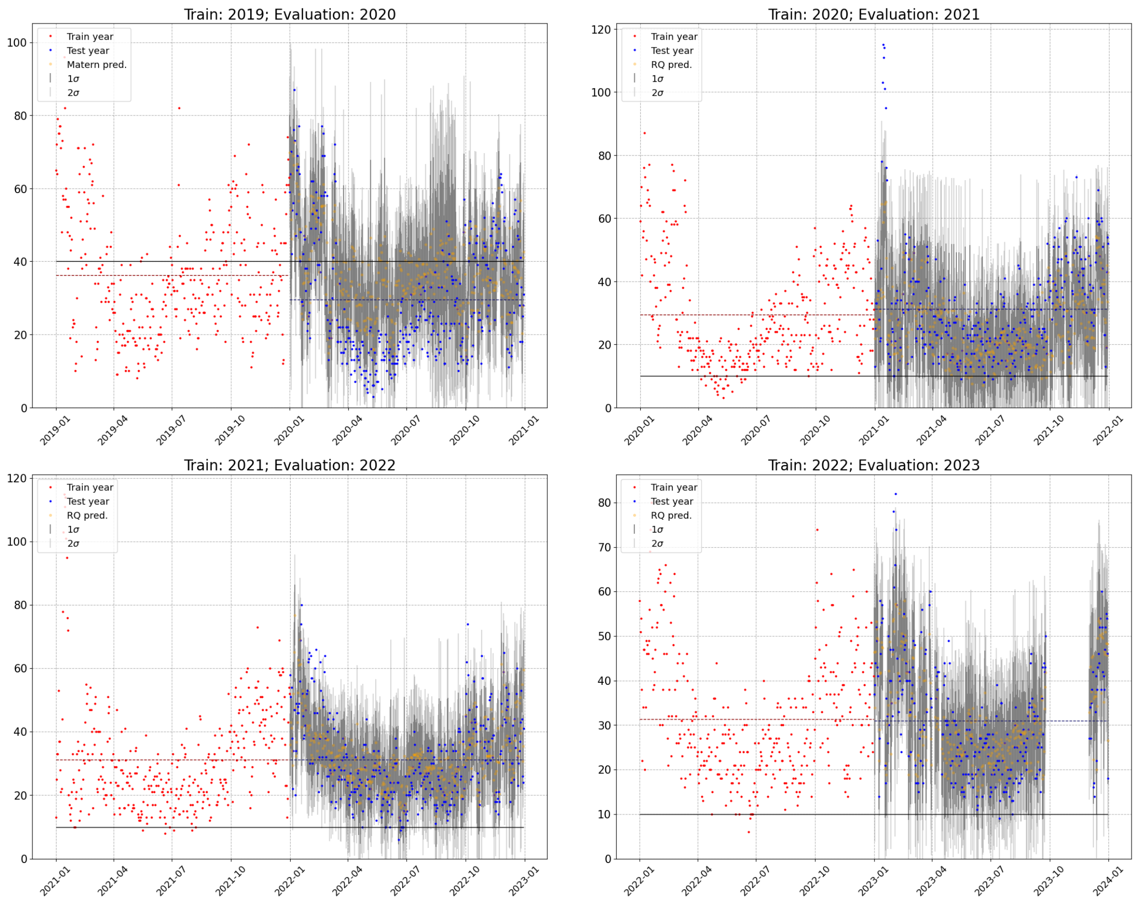


Fig. 4.2. Results of the GP predictions using the kernel that best predicts the evaluation year. The WHO recommended levels of NO_2 annual mean are shown in a solid black line. A dark red dashed line is used for the training year and a dark blue dashed line for the next year's observations. The error bars show the regions within one sigma in strong gray and a lighter gray represents the two sigmas region. Note that these figures are just a visualization of the accuracy of the models and how different seem to be the concentration of NO_2 from year to year. The x-axis is chosen just as a label to define the order of the observations since time does not represent a variable included in the models.

This behavior appears to be coherent with the implementation of MC but it also could be considered as a consequence of the COVID-19 pandemic. In that case, only the decrease in 2028 and 2019 could be understood as a direct consequence of the MC measure. However, 2019 is the first year where MC was fully implemented.

In 2021 the WHO guidelines were updated and the recommended levels of NO_2 were reduced. In Fig. 4.1 can be observed that Plaza del Carmen has struggled to comply with the WHO guidelines. This fact is more evident from 2015 to 2018 were the NO_2 were farther from the 40 μg/m^3 guideline and from 2021 onward with the 10 μg/m^3 guideline. Only in 2019 and 2020 were the WHO guidelines satisfied, and in 2013 and 2014 closer values to the WHO objective were measured.

To gain insight into this question, the explained GPRs are trained. It is expected that a more sophisticated tool could help in making inferences from the data. As commented, the proposed methodology aims to predict the data from one year using the previous one. If both years are distributed in a similar way the prediction error rate is expected to be low. Thus, to infer if MC had a significant impact, the GPR is expected to predict with a similar performance in terms of RMSE for all the years before the MC implementation. An increase of the RMSE should appear in the year that changes the distribution of the pollutant. From this breaking point onward the RMSE should yield values similar to those observed for the years before.

In Fig. 4.2 a visualization of the model outputs is displayed. These graphics represent a visualization of the model performance and do not try to show samples of the stochastic GP. Note that the role of the time in these figures is just a way to label the observation following a natural order established by choosing to train year by year.

The plotted figures show a solid black line which represents the recommended average concentration of NO_2 by year. The average mean observed for the training year is displayed as a red dotted line, whereas the observed evaluation year average of NO_2 is shown as a blue dotted line. The observed values of the response variable used for training are presented as red dots. The observed test values were labeled with blue whereas the predictions made for the model are illustrated as orange dots. Additionally, the 1σ and 2σ intervals are illustrated in different scales of gray as error bars. A stronger gray is chosen for the former and light gray for the latter.

Only GPRs are displayed in Fig. 4.2. The GP considered was only used between the two best candidates of structure models. Furthermore, only the GPR with the best predictions for each evaluation year is shown in the plot.

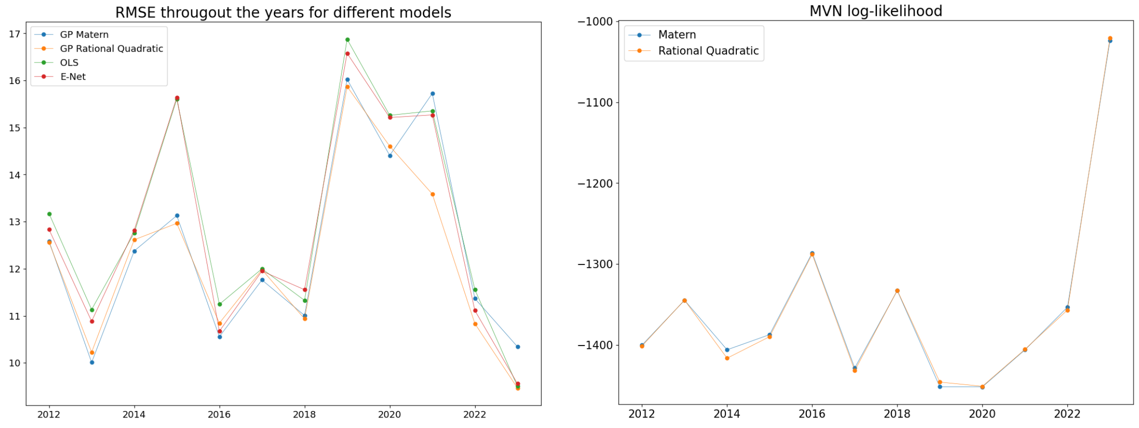
For Gaussian populations, it is well known that the 1σ and 2σ symmetric intervals represent a confidence level of 68% and 95% respectively. These intervals, however, have a considerable width, being one standard deviation around $\sim 10\ \mu g/m^3$ for each model.

The data before MC (i.e., the evaluation years between 2012–2017) seems to exhibit a good behavior of the predictions concerning the true observed values. All points seem to be close to the actual values. There are questionable cases, such as 2012 and 2014, where the orange dots (predictions) seem to be somewhat higher than the blue dots (real observations). The average mean of the NO_2 levels shows a relevant decrease from 2011 to 2012 and a notable increase from 2014 to 2015. The rest of the plots, until 2017, do not exhibit major decreases or increases in the average mean compared with the previous year and the performance of the models seems to be consistent. Regarding the WHO guideline of the 40 $\mu g/m^3$, none of these years seem to adhere to that value. However, in 2013 and 2014 the yearly mean levels of NO_2 were fairly close. Relevant issues with observed values leaving the 1σ and 2σ intervals include 2012, with some blue points under the 1σ interval (which could cause the decrease in annual mean compared with 2011) and a smaller amount of blue points in 2017 leaving the 1σ both upward and downward. Furthermore, for some outliers at the end of 2014 that surpasses the upper limit of the 2σ interval.

In the studied years affected by the MC implementation (2018–2023) two relevant decreases of the observed annual mean of NO_2 took place in 2019 and 2020. The predictions seem to match reasonably the observed test data. However, in 2019 and 2020 it seems that the predictions overestimate the values of the actual pollutant levels. This can be seen with the orange cloud of dots above the blue dots both in 2019 and 2020. In 2020 the separation between the predicted values and the actual observed values seems to be clearer. In both of these cases, it seems that a considerable amount of observed values abandon the 1σ interval below the inferior bound. It seems that some outliers have been detected outside the 2σ interval in 2019, but only in a few cases (both above and below the confidence interval). For 2020 this effect is less drastic, separating from the 1σ interval below the inferior more moderately. In these two years, the predictions seem to exhibit some bias, since the values expected for the GPR were lower than the actual values recorded. This contrasts with 2018, where some observed values (blue dots) outside 1σ span both below and above this interval. From 2021 onward no major changes in the yearly mean of pollutant concentration were detected. Besides the predicted values appear to be mixed with the blue cloud, with no indications of a clear bias on the models. However, some outliers appear at the beginning of 2021 outside of the 2σ interval.

4.1. Result of metrics

Many of the relevant results of this work are presented in Tab. 4.1 where the results of all the metrics are shown. Visualizations of these values are presented in Fig. 4.3, where the behavior of the metrics can be understood more intuitively.



(a) (b)

Fig. 4.3. Visualization of the results for the model performance. Both RMSE and log $\mathcal{L}$ for GPR are compared side by side.

As we can see the behavior of the log-likelihood in Fig. 4.3 (b) is pretty similar for both the rational quadratic kernel and the Matern $\nu = 3/2$ kernel. So in this application the use of the log-likelihood, although attractive given its clear interpretation, does not seem very insightful. Maybe different preprocessing of the data could help to make the data more sensible to this function or in other applications the differences could appear more clear. Such a sensible metric does not seem to be a good option when making inferences about the data for this case study.

On the other hand, the RMSE behavior reflected in Figure 4.3 exhibits clear differences between models, even within GPRs. On one hand, the linear models seem to share a similar behavior, while the GPRs appear to follow a pattern similar to each other but different from that of the linear models.

The OLS and elastic net share a pretty similar performance but in general, the performance of the elastic net is better. When the OLS surpasses the elastic net the difference in performance seems to be lesser than the occasions in which the elastic net performs OLS (the exception would be 2018). The overall performance of the elastic net reflects that this model appears to adapt better than OLS throughout the years as was expected. Both linear models exhibited a huge increase in the error rate in 2015 (when the drastic increase of NO_2 levels was produced), and 2019 (when a relevant decrease of NO_2 was observed). In 2020 (when the pandemic curfew took place) high values of the RMSE were recorded for the scale of recorded error rates, but there was a decrease respect 2019.

For the GPR the situation is not the same. Increments of NO_2 are indeed experienced in 2015 and 2019. However, 2015 does not represent a drastic peak of RMSE compared to other years such as 2012 or 2014, with comparable values (which was not the case for linear models).

Note that these facts do not imply that GPRs outperform the linear model in all years since elastic net can beat the Matern GPR from 2020 onward and also outperform the rational quadratic GPR in 2016. And OLS beats the Matern GPR in 2021 and 2023.

Ev. year	RMSE-M	$\log \mathcal{L}$-M	RMSE-RQ	$\log \mathcal{L}$-RQ	RMSE-OLS	RMSE-enet
2012	12.58	-1400.31	12.56	-1401.32	13.17	12.83
2013	10.01	-1344.80	10.23	-1344.67	11.13	10.88
2014	12.38	-1406.03	12.61	-1416.42	12.76	12.82
2015	13.13	-1387.31	12.96	-1389.84	15.61	15.64
2016	10.55	-1286.27	10.84	-1287.88	11.24	10.67
2017	11.76	-1428.51	11.98	-1431.77	12.00	11.95
2018	11.01	-1332.86	10.94	-1332.51	11.33	11.55
2019	16.02	-1451.77	15.87	-1445.96	16.88	16.58
2020	14.41	-1451.92	14.60	-1451.17	15.26	15.21
2021	15.72	-1406.13	13.59	-1405.53	15.35	15.27
2022	11.37	-1353.41	10.83	-1356.78	11.56	11.11
2023	10.34	-1023.54	9.47	-1020.42	9.50	9.56

Table 4.1. PERFORMANCE OF ALL THE MODEL CONSIDERED. MATERN AND RATIONAL QUADRATIC KERNEL ARE LABELED JUST AS “M” AND “RQ” FOR BREVITY.

These observed results lead us to focus on RMSE when making insights about the data. To check if the $\log \mathcal{L}$ is a relevant metric relative deviations are presented in Tab. 4.2 to compare how sensible each metric is in this case study.

The relative deviation percentage is computed to infer how far a value is from a reference value. Usually the reference value is a tabulated value or a value that has a special interest concerning the tested value. It is carried out as follows:

$$\text{Rel. dev.}(r_{\text{tested}}, r_{\text{reference}})\% = \frac{|r_{\text{tested}} - r_{\text{reference}}|}{|r_{\text{reference}}|} \times 100. \tag{4.1}$$

However, when comparing how far the results obtained from one GPR to another, neither of these models has an additional interest. Using the expression (4.1) could be useful if a model is accepted as the standard model used for a concrete application. But in this case, the following computation is suggested as a symmetric deviation percentage:

$$\text{Symm. dev.}(r_1, r_2)\% = \frac{|r_1 - r_2|}{|r_2|} \times 100. \tag{4.2}$$

In Tab. 4.2 are presented both computations to compare if the operation (4.2) is a compelling measure of deviation compared with the more commonly used (4.1). Looking at the results the $\log \mathcal{L}$ does not reflect the behavior desired since all deviations have values somewhat small for every year. For this application, the log-likelihood seems to lack sensibility.

	RMSE rel. dev. %			Log Marginal Likelihood rel. dev. %		
Ev. year	GPs	Matern w.r.t. RQ	RQ w.r.t. Matern	GPs	Matern w.r.t. RQ	RQ w.r.t. Matern
2012	0.20	0.20	0.20	0.07	0.07	0.07
2013	2.15	2.13	2.17	0.01	0.01	0.01
2014	1.90	1.88	1.92	0.74	0.73	0.74
2015	1.26	1.27	1.25	0.18	0.18	0.18
2016	2.70	2.67	2.74	0.13	0.12	0.13
2017	1.84	1.82	1.86	0.23	0.23	0.23
2018	0.65	0.65	0.64	0.03	0.03	0.03
2019	0.96	0.96	0.96	0.40	0.40	0.40
2020	1.36	1.35	1.37	0.05	0.05	0.05
2021	14.58	15.73	13.59	0.04	0.04	0.04
2022	4.80	4.91	4.68	0.25	0.25	0.25
2023	8.86	9.27	8.49	0.30	0.31	0.30

Table 4.2. RELATIVE DEVIATIONS IN PERCENTAGES BETWEEN THE GPRS FOR THE RMSE AND $\log\mathcal{L}$. IN THE COLUMN LABELED AS GPS SYMMETRIC RELATIVE DEVIATION IS USED.

In Tab. 4.3 relative deviations are used to compare the two types of GPR with OLS and elastic net. In this case, the linear models are used as benchmarks and therefore it seems fit to use them as reference values in (4.1). The symmetric deviation discussed before is presented again for the GPR and the $\log\mathcal{L}$ is discarded.

	RMSE OLS rel. dev. %		RMSE E-Net rel. dev. %		Symm. dev. %
Ev. y.\Rel. dev.	Matern	RQ	Matern	RQ	between GPs
2012	4.47	4.66	1.95	2.14	0.20
2013	10.10	8.15	8.04	6.05	2.15
2014	3.00	1.14	3.45	1.60	1.90
2015	15.88	16.93	16.06	17.11	1.26
2016	6.13	3.56	1.09	1.62	2.70
2017	1.98	0.16	1.58	0.25	1.84
2018	2.83	3.46	4.74	5.36	0.65
2019	5.05	5.95	3.34	4.26	0.96
2020	5.59	4.30	5.30	4.00	1.36
2021	2.41	11.51	2.99	11.01	14.58
2022	1.64	6.25	2.29	2.50	4.80
2023	8.85	0.39	8.22	0.97	8.86

Table 4.3. RELATIVE DEVIATION BETWEEN GPRS WITH RESPECT TO OLS AND E-NET. SYMMETRIC DEVIATION BETWEEN BOTH GPRS.

Finally in Tab. 4.4 the average of the RMSE from each type of GPR between 2012 and 2017 is computed. Then the relative deviation percentages concerning these values are computed for the years affected by MC.

	RMSE rel. dev. %	
Pre MC	Matern	RQ
mean	11.74	11.86
2018	6.22	7.83
2019	36.55	33.78
2020	22.76	23.09
2021	33.99	14.53
2022	3.14	8.68
2023	11.86	20.22

Table 4.4. RELATIVE DEVIATION FOR EACH YEAR AFFECTED BY MC WITH RESPECT TO THE AVERAGE VALUE OF THE RMSE BETWEEN 2012–2017.

4.2. Discussion of the results

In Fig. 4.2, it appears that a significant breaking point occurred in either 2019 or 2020, as the GPR model consistently predicted higher values than those observed in both years. Furthermore, numerous observations during these years fell below the 1σ confidence interval, unlike in other years. This suggests that the training data might not have been a representative sample of the evaluation population, introducing bias into the model. As a result, the RMSE values were higher, indicating a greater discrepancy between the predictions and the actual observations in those years compared to the rest of the study period.

Both the OLS and elastic net models, serving as benchmarks, were generally outperformed across the cases considered. This supports the GPR model as a strong candidate for this case study. However, neither GPR model was consistently the best-performing one in every year. In certain years, elastic net and OLS achieved lower RMSE values than one of the two GPR models. Nevertheless, in each year the best model is one of the two GPR. This is illustrated in Fig. 4.3 (a).

Furthermore, in Tab. 4.3, we can observe how the deviation in RMSE of GPRs from the linear models in 2015 reflects that GPRs can be more flexible. This differences are around ∼16–17%, the largest deviation observed in comparison to the linear models. In contrast, the two GPR models differed by only 1.26% in RMSE during that year.

When comparing the Matern GPR and the rational quadratic GPR, the performance of the Matern in terms of RMSE surpasses the rational quadratic before 2018 (except for 2015 when it seems to occur an unexpected increase of NO_2 levels). From 2018 onward, the rational quadratic outperforms the Matern kernel (except for 2020 with the COVID-19 pandemic). Furthermore, the Matern kernel performed particularly poorly from 2021 onward, being the worst model in 2021 and 2023, and it was outperformed by elastic net and the rational quadratic in 2022 (surpassing only OLS).

Regarding the deviation between GPRs in Tab. 4.3, the year 2021 was a notable year with the largest deviation between the two GPR models. This result might indicate that the wind data supply was not informative enough for the Matern kernel. The deviations from 2022 and 2023, while less pronounced, were still two to four times larger than the deviations seen in previous years.

These results suggest that assuming a GP with a Matern covariance is more appropriate for the years prior to 2018 and the rational quadratic GP is a better assumption afterwards. Thus, 2018 could also be considered a potential breaking point. The rational quadratic is particularly suitable for modeling NO_2 concentration after two major events had happened: the MC implementation and the global pandemic.

In order to quantify the differences between predictions before and after the implementation of the LEZ, refer to Tab. 4.4. Relative deviations are computed with respect to the average RMSE values for the six years preceding the MC implementation for each GPR model. This allows a clearer comparison of NO_2 levels before and after the LEZ activation. In this table, 2019 stands out as the year with the greatest change in RMSE relative to the years before MC, higher than 2020 and 2021. Both GPR models exhibit this significant shift in RMSE, despite the Matern kernel having an overall worse performance in the later years.

The change in model performance in 2018 appears minimal compared to other potential breaking points. The high RMSE values in 2020 can be attributed primarily to the COVID-19 curfews. In 2021, the Matern kernel's error increased, possibly due to the use of wind data from a different station, which might have compromised accuracy. Meanwhile, the rational quadratic GPR and linear models achieved lower RMSE values in this year than in 2020. It is unclear whether this discrepancy is due to using wind data from the Cuatro Vientos station instead of Retiro Park or if the Matern kernel lacked the flexibility required to adapt to post-MC implementation and pandemic conditions. It is also worth noting that although Tab. 4.4, shows a substantial difference between the performance of 2023 and the years previous to MC, this relative deviation is caused by a decrease in RMSE and not by an increase, as it can be seen in Fig. 4.3 (a). This indicates that the training data used in 2022 was informative for predicting the NO_2 levels in 2023.

Additionally, the results discussed in this section provide strong evidence that 2019 represents a year when a significant change occurred in the statistical distribution of NO_2 concentrations. This year also marks the only time that the Plaza del Carmen station adhered to the WHO guidelines established at that time. The impact of the MC initiative became notably significant from 2019, when it was fully implemented, rather than in 2018. Following this breaking point, NO_2 levels were further reduced by the pandemic-induced curfews and have remained relatively stable through 2023.

5. CONCLUSIONS AND FUTURE WORK

After studying the relevant theoretical foundation, a proposed methodology is presented to analyze the magnitude of the impact of a LEZ. Meteorological data is collected from the Retiro Park station from the AEMET OpenData API to be used as covariates. The NO_2 concentration levels data is taken from available pollutant data from the Plaza del Carmen station from the Madrid City Council. The lack of wind-related magnitudes in 2021 was supplied with data from the Cuatro Vientos station.

Then different models were evaluated to compare with the GPR modeled with a Matern $\nu = 3/2$ kernel and the GPR modeled with a rational quadratic kernel. This comparison was based on the predictive ability of RMSE.

The analysis proposed to evaluate a LEZ is to predict values of the pollutant levels taking into account the meteorological conditions since the wind and rain are known as natural pollutant dispersion mechanisms. Information from the previous year is used for training the models. Therefore, the proposed methodology would classify the impact of an event as relevant if the RMSE values of an accurate model for the application experiment increase from one year to another. Thus reflecting that the assumed distribution for the training year has experimented with a relevant change concerning the evaluation year. Then is reasonable to think that different assumptions have to be made for the years before and after the breaking event.

To consider if the selected GP structures represent accurate models, the performance of OLS and elastic net are used as references.

5.1. Future work

The results obtained in this work show consistency with the expected behavior for classifying a LEZ as either effective or insufficient. The application of the proposed methodology to other LEZs can reflect not only the impact of the zones themselves but also the robustness and variability of the results across different settings.

The imputation of missing values could be revisited. Exploring alternative imputation methods for different data types, along with a more thorough selection of hyperparameters for these methods, could significantly enhance the performance of GPs.

Additionally, hypothesis testing and goodness-of-fit assessments for GPs could be conducted. These tools would provide a more transparent interpretation of the results and potentially reduce the need to evaluate multiple years. Instead of relying solely on contextualizing and comparing results based on performance metrics, the impact of the LEZ could be evaluated in terms of statistical significance across different years.

5.2. Further issues

In this study, only GPs with zero mean are considered. In Sec. 2.7 from [1] is illustrated how modeling a relevant prior mean function for the application has interesting effects. This is typically achieved by specifying an explicit set of basis functions with coefficients that need to be determined during the training phase. Expert knowledge in the relevant domain is essential for determining the appropriate basis functions for the prior mean. In this framework, the GP could be used to fit the model noise, while the mean function establishes a more specific relationship between predictors and response variables.

Moreover, expert knowledge could also aid in selecting the most relevant predictors for the GPR. In this work, the criterion for predictor selection was to include all available data from AEMET, with the goal of providing the GP with the maximum possible information.

For more advanced modeling of covariance functions, the thesis by [12] appears to be a valuable reference. A more meticulous design of custom kernels could be pursued to capture distinct behaviors or trends in the data. However, increasing the number of hyperparameters in the model would also increase the computational cost of optimization.

Furthermore, no consideration about the implementation of categorical variables has been considered. A possible starting point for handling mixed data is to replace the Euclidean norm in isotropic kernels with similarity-based distances. An example is the Gower coefficient, which is well-suited for handling mixed data types. These types of distances are commonly used in multidimensional scaling. Proving the positive semi-definiteness of these newly defined kernels would be a necessary step.

The implementation of more complex kernels could be explored by combining existing kernels in Scikit-learn, using the properties discussed in Sec. 2.3.2. However, it should be noted that Scikit-learn does not support the introduction of custom kernels or non-zero prior mean functions.

5.3. Conclusions

The observed stabilization of NO_2 levels after 2019 reflects the combined impact of policy measures such as the MC initiative and external factors like the COVID-19 pandemic. While 2019 seems to be the most critical year in terms of statistical shifts, the continuation of these trends through 2023 suggests that both policy interventions and behavioral changes resulting from the pandemic had long-term effects on air pollution levels.

Further, the comparison of GPR models to the linear models (OLS and Elastic Net) consistently showed that GPR models are more effective in capturing the complexity of

NO_2 dynamics, especially when nonlinear relationships between meteorological variables and pollution levels are at play. However, the choice of kernel within the GPR framework remains crucial, as the Matern kernel underperformed in later years, particularly after 2018. The rational quadratic kernel, on the other hand, showed greater adaptability post-2018, except for anomalies caused by external events such as the pandemic in 2020.

Looking at the broader implications of this study, the application of GPR models in environmental forecasting can help policymakers better understand and anticipate shifts in pollutant levels. By identifying breaking points like 2019, cities can implement targeted interventions aimed at sustaining improvements in air quality, ensuring adherence to public health guidelines such as those set by the WHO.

In conclusion, while GPR models are computationally intensive, their ability to model complex, nonlinear relationships makes them valuable tools in environmental studies. This is particularly important when analyzing the effects of large-scale interventions, such as the implementation of LEZs. This study further supports the notion that the rational quadratic GPR is the preferred model for post-2018 data, and that the breaking point in NO_2 levels occurred in 2019, likely due to a combination of policy implementation and global events.

Therefore the methodology applied in this work classifies the impact of the case study, MC as relevant, since the breaking point in the distribution of pollutant levels occurred in 2019 when the change in the distribution produce a decrease pollutant levels with remarkable relative deviations from the previous distribution. The influence of MC is reflected since 2019 was the first year when MC was fully implemented and was not influenced by pandemic-related restrictions. Furthermore, the effects are sustained over time, with a low prediction error observed in the later years (2022-2023), when predictor data was not missing and after the global pandemic curfew ended.

APPENDIX

- Tuned hyperparameters obtained for each GPR. The year shown represents the training year:

```
2010  : 52.5**2 * Matern(length_scale=18.6, nu=1.5) + WhiteKernel(noise_level=123)
2011  : 55.8**2 * Matern(length_scale=17.7, nu=1.5) + WhiteKernel(noise_level=127)
2012  : 56.9**2 * Matern(length_scale=17.6, nu=1.5) + WhiteKernel(noise_level=91.7)
2013  : 27.4**2 * Matern(length_scale=8.04, nu=1.5) + WhiteKernel(noise_level=68.9)
2014  : 46.6**2 * Matern(length_scale=12.3, nu=1.5) + WhiteKernel(noise_level=84.3)
2015  : 62.3**2 * Matern(length_scale=19.4, nu=1.5) + WhiteKernel(noise_level=100)
2016  : 44**2 * Matern(length_scale=18.2, nu=1.5) + WhiteKernel(noise_level=74.6)
2017  : 35.4**2 * Matern(length_scale=12.4, nu=1.5) + WhiteKernel(noise_level=110)
2018  : 47.1**2 * Matern(length_scale=22.1, nu=1.5) + WhiteKernel(noise_level=93.7)
2019  : 42.8**2 * Matern(length_scale=16.3, nu=1.5) + WhiteKernel(noise_level=96.9)
2020  : 28.8**2 * Matern(length_scale=8.81, nu=1.5) + WhiteKernel(noise_level=97.7)
2021  : 34**2 * Matern(length_scale=9.07, nu=1.5) + WhiteKernel(noise_level=73.5)
2022  : 26**2 * Matern(length_scale=10.2, nu=1.5) + WhiteKernel(noise_level=78.4)

2010  : 68.6**2 * RationalQuadratic(alpha=0.0312, length_scale=16.8) + WhiteKernel(noise_level=121)
2011  : 67.7**2 * RationalQuadratic(alpha=0.0773, length_scale=15.1) + WhiteKernel(noise_level=129)
2012  : 57.6**2 * RationalQuadratic(alpha=0.0179, length_scale=10) + WhiteKernel(noise_level=80.1)
2013  : 47.5**2 * RationalQuadratic(alpha=0.0205, length_scale=6.62) + WhiteKernel(noise_level=52.4)
2014  : 64.7**2 * RationalQuadratic(alpha=0.0349, length_scale=11.2) + WhiteKernel(noise_level=79.7)
2015  : 76.9**2 * RationalQuadratic(alpha=0.0646, length_scale=17.2) + WhiteKernel(noise_level=102)
2016  : 57.1**2 * RationalQuadratic(alpha=0.0134, length_scale=10.9) + WhiteKernel(noise_level=60.4)
2017  : 55.3**2 * RationalQuadratic(alpha=0.0176, length_scale=10.3) + WhiteKernel(noise_level=98)
2018  : 52.8**2 * RationalQuadratic(alpha=0.0202, length_scale=12.1) + WhiteKernel(noise_level=83.2)
2019  : 54.1**2 * RationalQuadratic(alpha=0.0708, length_scale=14.2) + WhiteKernel(noise_level=97.6)
2020  : 39.1**2 * RationalQuadratic(alpha=0.0685, length_scale=7.68) + WhiteKernel(noise_level=94.8)
2021  : 47.4**2 * RationalQuadratic(alpha=0.0279, length_scale=4.88) + WhiteKernel(noise_level=39.2)
2022  : 40.9**2 * RationalQuadratic(alpha=0.0209, length_scale=7.76) + WhiteKernel(noise_level=67.4)
```

BIBLIOGRAPHY

[1] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[2] J. L. Gómez-González and M. Cárdenas-Montes, "Gaussian process-based analysis of the nitrogen dioxide at Madrid Central Low Emission Zone," *Logic Journal of the IGPL*, vol. 32, no. 4, pp. 700–711, Apr. 2024.

[3] W. H. Organization, *Air quality guidelines: global update 2005: particulate matter, ozone, nitrogen dioxide, and sulfur dioxide*. World Health Organization, 2006.

[4] W. H. Organization *et al.*, *WHO global air quality guidelines: particulate matter (PM2. 5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization, 2021.

[5] K. Murphy, *Machine Learning: a probabilistic perspective*. MIT press, 2012.

[6] K. P. Murphy, *Probabilistic machine learning: an introduction*. MIT press, 2022.

[7] J. Wang, "An intuitive tutorial to gaussian processes regression," *Computing in Science & Engineering*, 2023.

[8] C. Williams and C. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8, MIT Press, 1995.

[9] I. Molina Peralta and E. García-Portugués, *A First Course on Statistical Inference*. 2024, Version 2.4.1. ISBN 978-84-09-29680-4.

[10] Y. L. Tong, *The multivariate normal distribution*. Springer Science & Business Media, 2012, ch. 3.

[11] E. García-Portugués, *Notes for Nonparametric Statistics*. 2023, pp. 28–29, Version 6.9.0. ISBN 978-84-09-29537-1.

[12] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, Apollo - University of Cambridge Repository, 2014.

[13] D. J. MacKay *et al.*, "Introduction to gaussian processes," *NATO ASI series F computer and systems sciences*, vol. 168, pp. 133–166, 1998.

[14] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[16] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.