

André Igor Nóbrega da Silva

**Estudo da Regressão por Processo Gaussiano
para Previsão da Evolução dos Casos de
Covid-19 em Campina Grande - PB**

Campina Grande, Paraíba

Outubro de 2021

André Igor Nóbrega da Silva

**Estudo da Regressão por Processo Gaussiano para
Previsão da Evolução dos Casos de Covid-19 em
Campina Grande - PB**

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Engenharia Elétrica – DEE

Orientador: Edmar Candeia Gurjao, D. Sc.

Campina Grande, Paraíba
Outubro de 2021

André Igor Nóbrega da Silva

Estudo da Regressão por Processo Gaussiano para Previsão da Evolução dos Casos de Covid-19 em Campina Grande - PB

Trabalho de Conclusão de Curso submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Paraíba, 14 de dezembro de 2020:

Edmar Candeia Gurjao, D. Sc.
Orientador

Edson Porto da Silva, D. Sc.
Convidado

Campina Grande, Paraíba
Outubro de 2021

À minha família, Solange Nóbrega, José Artur e Arthur Victor por me trazerem até aqui.

Agradecimentos

Ao longo do desenvolvimento deste trabalho, recebi bastante ajuda e suporte.

Primeiramente gostaria de agradecer ao meu orientador, Edmar Candeia, pela oportunidade de desenvolver esse trabalho e pelo acompanhamento cuidadoso e constantes ensinamentos nesses últimos meses. Gostaria de agradecer, também, à professora Elloá Guedes, cuja experiência foi essencial na estimulação do pensamento crítico e na definição de uma metodologia de pesquisa. Tais orientações elevaram o nível deste trabalho e me ajudaram a melhorar na pesquisa científica.

Além disso gostaria de agradecer aos amigos que ganhei ao longo da graduação, que formaram uma segunda família para mim fora de casa e que me ensinaram mais do que a sala de aula. Em especial, agradeço à minha melhor amiga e namorada Camila Machado, pelo carinho, pelos ensinamentos, pelo amor e pelo apoio incondicional. À Maxwell Albuquerque, Pedro Enio, Ana Paula e Marianna Barbosa pelos diversos trabalhos e desafios que enfrentamos juntos. Com vocês os momentos mais difíceis se transformaram nos melhores. Agradeço à Cláudio Lisboa por me incentivar à cursar Engenharia Elétrica e pela amizade nas situações mais difíceis. À Saulo Sobreira, Pedro Ximenes, Mário Cesar, Thiago Delmiro, Rodrigo Kalil, Matheus Rocha, Arthur Dimitri, Valmir pelas ajudas, pelo companheirismo, pelas alegrias, felicidades e dificuldades ao longo desses anos.

Por fim, gostaria de agradecer especialmente à minha família, Solange Nóbrega, José Artur e Arthur Victor por me trazerem até aqui. Vocês sempre estiveram lá por mim nos conselhos sábios, no entendimento e na torcida pelo sucesso.

Know how to solve every problem that has ever been solved. - Richard Feynman

Resumo

As expectativas atuais acerca da Inteligência Artificial e da Aprendizagem de Máquina são grandes — tanto na comunidade científica quanto no mercado. A ideia de que podemos extrair funções complexas a partir de uma grande quantidade de dados é inspiradora e nos leva a acreditar que podemos melhorar nossa capacidade de predição de eventos futuros e de tomada de decisões a partir de uma metodologia baseada em dados. Podemos utilizar o do Processo de Regressão Gaussiano para realizar tarefas de aprendizagem supervisionada em séries temporais. Trata-se de um algoritmo de regressão não linear e não paramétrico baseado na aprendizagem bayesiana. Neste trabalho realizamos um estudo acerca das principais características do Processo de Regressão Gaussiano com foco nas características de suas funções *kernel*. São analisados 3 problemas básicos que nos permite visualizar as etapas de treinamento do algoritmo e as propriedades das tais funções *kernel*. Em seguida, realizamos uma aplicação prática para a previsão da evolução dos casos de Covid-19 na cidade de Campina Grande - PB.

Palavras-chave: Inteligência Artificial, IA, Aprendizagem de Máquina, Processo de Regressão Gaussiano, GPR, Aprendizagem Bayesiana, Bayes, Covid-19, Séries Temporais, Campina Grande.

Abstract

Current expectations involving Artificial Intelligence and Machine Learning are high — both in the scientific community and in the industry. The idea that we can extract complex functions from a large amount of data is inspiring and leads us to believe that we can improve our ability to predict future events and make better decisions using a data-driven methodology. Within this context, we can use the Gaussian Regression Process algorithm to perform supervised learning tasks in time series. It is a non-linear and non-parametric regression algorithm based on Bayesian learning. In this we study the main characteristics of the Gaussian Regression Process, focusing on the properties of its *kernel* functions. 3 basic problems are analyzed allowing us to visualize the algorithm training steps and the distinctions between such *kernel* functions. Then, we apply it in practical problem to predict the evolution of Covid-19 cases in the city of Campina Grande - PB.

Keywords: Artificial Intelligence, AI, Machine Learning, ML, Gaussian Regression Process, GPR, Bayesian Learning, Bayes, Covid-19, Time Series, Campina Grande.

Lista de ilustrações

Figura 1 – Distribuição a priori da classe ovelhas	18
Figura 2 – Distribuição posteriori após o aprendizado	18
Figura 3 – Dados de treinamento para a regressão linear	20
Figura 4 – Inferência utilizando uma distribuição priori fraca	21
Figura 5 – Inferência utilizando uma distribuição priori forte	22
Figura 6 – Exemplos de funções Kernel	24
Figura 7 – Amostras a priori e a posteriori	25
Figura 8 – Funções <i>kernel</i> base	27
Figura 9 – Exemplo de treinamento com função <i>kernel</i> linear de expoente 2	29
Figura 10 – Combinações de funções <i>kernel</i> base por meio de soma	29
Figura 11 – Combinações de funções <i>kernel</i> base por meio de multiplicação	30
Figura 12 – Exemplos de séries temporais com diferentes componentes	33
Figura 13 – Ilustração da validação cruzada para séries temporais	35
Figura 14 – Decomposição aditiva da série temporal do sinal $y_{sazonal}$	37
Figura 15 – Decomposição aditiva da série temporal do sinal $y_{tendencia}$	39
Figura 16 – Decomposição aditiva da série temporal do sinal $y_{saz+tend}$	40
Figura 17 – Dados brutos de Covid em Campina Grande	41
Figura 18 – <i>Pipeline</i> de transformação dos dados brutos	42
Figura 19 – Mapa de calor da quantidade acumulada de casos por bairro em Campina Grande no início de Junho de 2020	43
Figura 20 – Mapa de calor da quantidade ativa de casos por bairro em Campina Grande no início de Junho de 2020	43
Figura 21 – Evolução do acumulado de casos de Covid em Campina Grande - PB	44
Figura 22 – Decomposição aditiva da série temporal do acumulado de casos	45
Figura 23 – Autocorrelação e autocorrelação parcial série temporal do acumulado de casos	46
Figura 24 – Modelagem de variáveis utilizadas para o acumulado de casos	46
Figura 25 – Evolução dos casos diários	48
Figura 26 – Decomposição aditiva da série temporal dos casos diários	50
Figura 27 – Autocorrelação e autocorrelação parcial série temporal da média móvel dos casos diários	51
Figura 28 – Amostras a priori e a posteriori do <i>kernel</i> periódico para a série sazonal	53
Figura 29 – Amostras a priori e a posteriori do <i>kernel</i> SE para a série sazonal	54
Figura 30 – Amostras a priori e a posteriori do <i>kernel</i> linear para a série sazonal	54
Figura 31 – Comparação do R^2 ao longo dos splits - apenas sazonalidade	55
Figura 32 – Comparação do MSE ao longo dos splits - apenas sazonalidade	56

Figura 33 – Comparação do MAPE ao longo dos splits - apenas sazonalidade	56
Figura 34 – Amostras a priori e a posteriori do <i>kernel</i> periódico para a série com tendência	57
Figura 35 – Amostras a priori e a posteriori do <i>kernel</i> SE para a série com tendência	58
Figura 36 – Amostras a priori e a posteriori do <i>kernel</i> linear para a série com tendência	58
Figura 37 – Comparação do R^2 ao longo dos splits - apenas tendência	59
Figura 38 – Comparação do MSE ao longo dos splits - apenas tendência	60
Figura 39 – Comparação do MAPE ao longo dos splits - apenas tendência	60
Figura 40 – Amostras a priori e a posteriori do <i>kernel</i> periódico para a série com tendência e sazonalidade	61
Figura 41 – Amostras a priori e a posteriori do <i>kernel</i> SE para a série com tendência e sazonalidade	62
Figura 42 – Amostras a priori e a posteriori do <i>kernel</i> linear para a série com tendência e sazonalidade	62
Figura 43 – Amostras a priori e a posteriori do <i>kernel</i> linear + periódico para a série com tendência e sazonalidade	63
Figura 44 – Comparação do R^2 ao longo dos splits - tendência e sazonalidade	64
Figura 45 – Comparação do MSE ao longo dos splits - tendência e sazonalidade . . .	64
Figura 46 – Comparação do MAPE ao longo dos splits - tendência e sazonalidade . .	65
Figura 47 – Comparação do R^2 ao longo dos splits - acumulado de casos	66
Figura 48 – Comparação do MSE ao longo dos splits - acumulado de casos	66
Figura 49 – Comparação do MAPE ao longo dos splits - acumulado de casos	67
Figura 50 – Predições e dados reais para o todo o conjunto de dados de casos acumulados	67
Figura 51 – Predições e dados reais para o último <i>split</i> de teste	68
Figura 52 – Erros do modelo para o último <i>split</i> de teste	68
Figura 53 – Comparação do R^2 ao longo dos splits - média móvel de casos	69
Figura 54 – Comparação do MSE ao longo dos splits - média móvel de casos	70
Figura 55 – Comparação do MAPE ao longo dos splits - média móvel de casos	70
Figura 56 – Predições e dados reais para o todo o conjunto de dados de média móvel	71
Figura 57 – Predições e dados reais para o último <i>split</i> de teste de média móvel . .	71

Lista de abreviaturas e siglas

GPR	<i>Gaussian Process Regression</i>
SE	Squared Exponential
RBF	Radial Basis Function
PB	Parábola
GP	<i>Gaussian Process</i>
RQ	<i>Rational Quadratic</i>
MSE	<i>Mean Squared Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
ts	<i>Time Series</i>
WN	<i>White Noise</i>

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.2	Estrutura do relatório	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Aprendizagem Bayesiana	16
2.1.1	Exemplo com Regressão Linear	18
2.1.2	Gerando previsões a partir do modelo treinado	22
2.2	Método de Regressão por Processos Gaussianos	23
2.2.1	Funções Kernel	26
2.2.1.1	<i>Kernel Squared Exponential (SE)</i>	27
2.2.1.2	<i>Kernel Periódico</i>	27
2.2.1.3	<i>Kernel Linear</i>	28
2.2.2	Estruturas de Kernels	28
2.3	Séries Temporais	31
2.3.1	Métricas de avaliação de previsões	33
3	METODOLOGIA	36
3.1	Série temporal apenas com componente de sazonalidade	36
3.1.1	Dados utilizados	36
3.1.2	<i>Kernels</i> propostos	37
3.2	Série temporal apenas com componente de tendência	38
3.2.1	Dados Utilizados	38
3.2.2	<i>kernels</i> propostos	38
3.3	Série temporal com componente de tendência e de sazonalidade	39
3.3.1	Dados Utilizados	39
3.3.2	<i>Kernels</i> Propostos	40
3.4	Previsão do acumulado de casos de Covid	41
3.4.1	Modelagem do problema	44
3.4.2	Estruturas de <i>kernels</i> proposta	46
3.5	Previsão da média móvel de casos de Covid	48
3.5.1	Modelagem do problema	49
3.5.2	Estruturas de <i>kernels</i> proposta	50
4	RESULTADOS OBTIDOS	53
4.1	Série temporal apenas com componente de sazonalidade	53

4.2	Série temporal apenas com componente de tendência	57
4.3	Série temporal com componente de tendência e de sazonalidade . .	61
4.4	Previsão do acumulado de casos de Covid	63
4.5	Previsão da média móvel de casos de Covid	68
5	CONCLUSÃO	73
	REFERÊNCIAS	75

1 Introdução

Ao longo do século XX, a ciência entrou em uma era de desenvolvimento da área da Inteligência Artificial (IA), que possui como objetivo construir agentes inteligentes capazes de aprender e de realizar suposições acerca do mundo que os engloba. Além das grandes aplicações práticas que tal tecnologia pode nos proporcionar, é válido destacar que trata-se de um ramo que, assim como a filosofia e a psicologia, também tenta explicar as sutilezas da inteligência humana.

As expectativas atuais acerca da IA e mais especificamente da Aprendizagem de Máquina são grandes. De acordo com o autor e pesquisador americano na área de estudos futuros, John Naisbitt, "*We are drowning in information and starving for knowledge*". De fato, estamos na era do Big Data, em que dados são gerados em uma quantidade tão grande e tão complexa que não é possível utilizar os paradigmas tradicionais da computação para extrair conhecimento e realizar previsões. É dentro desse contexto que torna-se importante a construção de métodos automáticos de análise de dados e de algoritmos capazes de capturar padrões em conjuntos de dados. Essas funções podem então serem utilizadas para realizar previsões em dados não antes vistos.

Dentro de uma grande classe de problemas que podem ser resolvidos de acordo com esse paradigma, podemos destacar a Aprendizagem Supervisionada, que é o foco deste trabalho. Trata-se do problema de encontrar uma função que mapeia entradas e saídas a partir de um conjunto de dados de treinamento. Ou seja, dado um conjunto $D = \{(x_i, y_i) | i = 1, 2, \dots, n\}$, em que x é a variável de entrada e y é a saída (alvo da previsão), queremos encontrar uma função f tal que $f(x) \approx y$. A partir desse processo, queremos utilizar a função f para realizar previsões em dados de entrada não antes vistos x_* , obtendo y_* . Como estamos saindo de um conjunto D limitado e utilizando a função aprendida para novos dados, podemos afirmar que o treinamento é um processo de indução (RASMUSSEN; WILLIAMS, 2006).

Existem diversos algoritmos e estratégias para encontrar tal função f de acordo com o tipo específico do problema proposto. Dentre tais, podemos destacar duas abordagens comuns (RASMUSSEN; WILLIAMS, 2006). Na primeira, fazemos considerações iniciais acerca da classe e da complexidade das possíveis funções. Por exemplo, podemos supor que uma função polinomial quadrática modela de maneira satisfatória as entradas às saídas. Uma dificuldade imediata que surge a partir disso é a decisão acerca da riqueza da função considerada. Caso haja pouca complexidade, pode-se surgir o problema do *underfitting*. Caso seja demasiadamente complexa, pode-se ter o *overfitting*. A outra abordagem propõe definir probabilidades prioris para cada classe de função possível e utilizar atualizações

Bayesianas no processo de aprendizagem.

É nessa segunda abordagem que o a Regressão por Processo Gaussiano (do inglês *Gaussian Regression Process*, ou *GPR*) se encaixa, já que é um método não-paramétrico que leva em consideração uma classe de funções que se ajustam aos dados e nos retorna, como predição, uma distribuição de probabilidade Gaussiana, definida por uma média e por um intervalo de confiança. Esse método vem se tornando uma alternativa popular a algoritmos como as Redes Neurais, já que incorpora uma natureza de incerteza às predições, além de apresentar bons resultados em bases de dados pequenas. Uma das vantagens da Regressão por Processo Gaussiano é que pode-se focar em reduzir a incerteza de maneira localizada ao longo da distribuição de dados, dando mais ou menos importância a certos aspectos e aumentando a explicabilidade do modelo.

Dentre os principais desafios na utilização do GPR podemos citar a dificuldade de entendimento do funcionamento do modelo e a dificuldade de encontrar uma função de covariância (ou *kernel*) adequada para o problema proposto (FREITAS, 2013). Neste trabalho realizamos um estudo acerca das principais características da Regressão por Processo Gaussiano, bem como apontamos os principais conceitos necessários para o seu entendimento. Propõe-se 3 problemas básicos iniciais (*toy problems*), adaptados de (ORDUZ, 2019) para a visualização do processo de treinamento do modelo e do impacto da utilização de diferentes funções *kernel*. Em seguida, partimos para a aplicação da teoria discutida, prevendo a quantidade de casos acumulados e a média móvel de casos de Covid-19 na cidade de Campina Grande - PB. São levantadas hipóteses acerca das estruturas de *kernel* que podem se adequar aos problemas.

1.1 Objetivos

Realizar um estudo do GPR, destacando o impacto das estruturas de funções *kernel* utilizadas para cada problema específico. Para isso, utilizamos 3 problemas básicos iniciais e, em seguida, fazemos uma aplicação prática de previsões da evolução de casos de Covid-19 na cidade de Campina Grande - PB. Os dados utilizados são fornecidos pela Secretaria de Saúde do município.

1.2 Estrutura do relatório

Este documento está dividido em 6 capítulos. O primeiro é dedicado à introdução e apresentação dos objetivos do trabalho. No segundo capítulo é apresentada a fundamentação teórica, que consiste nos princípios da aprendizagem Bayesiana, na apresentação do método GPR e nas métricas de avaliação de séries temporais. Segue-se, no terceiro capítulo, a metodologia de hipóteses e avaliações proposta para cada um dos problemas apresentados.

No quarto capítulo são apresentados os resultados obtidos nas métricas de avaliação. Por fim, no quinto capítulo traz-se as conclusões do trabalho.

2 Fundamentação Teórica

Neste capítulo serão apresentados os principais conceitos e processos envolvidos no treinamento e na realização de previsões utilizando a Regressão por Processo Gaussiano.

2.1 Aprendizagem Bayesiana

Na tarefa da aprendizagem supervisionada, buscamos utilizar um conjunto de treinamento $D = \{(x_i, y_i) | i = 1, 2, \dots, n\}$ para encontrar uma função f tal que $f(x) \approx y$. Esse é um problema específico dentro do campo da Aprendizagem de Máquina, que possui como propósito geral desenvolver agentes inteligentes capazes de aprender com experiências passadas. O objetivo dessa aprendizagem é de não somente utilizar o conhecimento para realizar ações em situações presentes, mas também realizar ações no futuro (RUSSEN; NORVIG, 1995).

Diversos algoritmos são propostos para a realização de tal atividade, e embora existam diferenças entre tais implementações, muitos desses métodos acabam executando uma inferência estatística.

Existem dois métodos principais para a realização da inferência estatística (CAUSEVIC, 2020): a abordagem frequentista e a abordagem Bayesiana. A primeira assume que os dados observados são advindos de alguma distribuição existente e utiliza-se algoritmos de otimização, como a minimização com o método dos mínimos quadrados ou a maximização do log-verossimilhança para encontrar tal função geradora. É importante apontar que existe, também, um certo parâmetro de incerteza associado aos dados de saída, ou seja, um erro inerente às observações. Dessa forma, tomando como exemplo a regressão linear, teríamos:

$$y_i = f(x_i) + \epsilon_i = \theta \times x_i + \epsilon_i$$

em que ϵ_i é um ruído modelado por uma distribuição normal com média 0 e variância σ^2 :

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

A tarefa torna-se, então encontrar um único parâmetro θ ótimo que maximize o ajuste aos dados.

A abordagem Bayesiana, por sua vez, associa a questão da incerteza não somente aos dados, mas também ao próprio parâmetro θ . Nesse método, utiliza-se o teorema de

Bayes para realizar inferências sobre distribuições e atualizar a certeza sobre os dados à luz de novas informações. A principal razão para a utilização dessa perspectiva é que o aprendizado ocorre de maneira incremental: a cada novo dado de entrada, atualizamos as distribuições de probabilidade de maneira adequada. Dessa forma, em vez de encontrar um único valor de θ que realiza um ajuste aos dados, encontramos uma distribuição de probabilidade de θ , construída a partir do exemplo de treinamento. As predições são realizadas levando em consideração todos os valores possíveis de θ , ponderados de acordo com a distribuição posteriori.

Conforme o nome sugere, a aprendizagem Bayesiana é fundamentada no teorema de Bayes. Esse teorema descreve a probabilidade de um evento baseado em um conhecimento a priori que pode estar relacionado ao evento. O teorema mostra como alterar as probabilidades a priori tendo em vista novas evidências para obter probabilidades a posteriori. Ou seja,

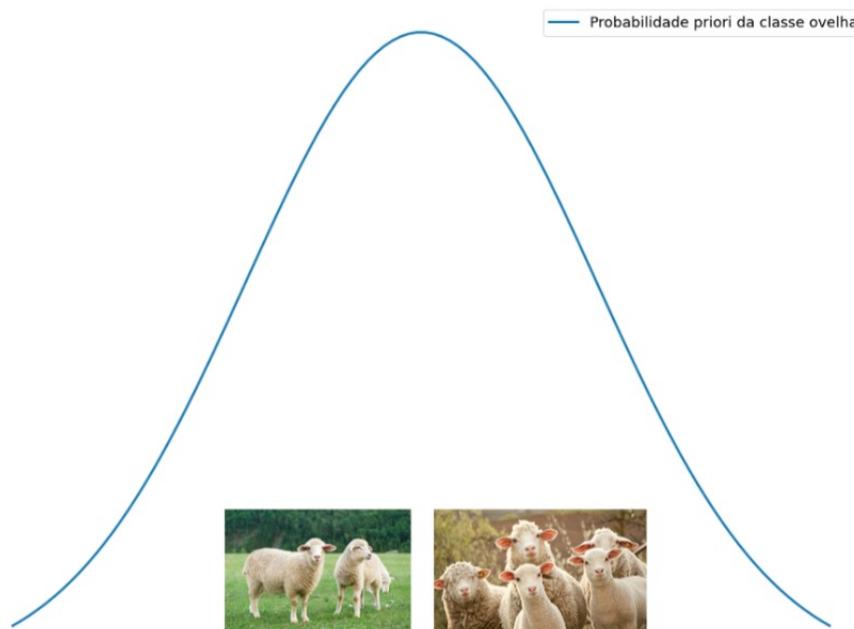
$$p(h|d) = \frac{p(d|h) \times p(h)}{\int_{h'} p(d|h') \times p(h') dh'} \quad (2.1)$$

Nesse caso, temos:

- $p(h|d)$: probabilidade da hipótese gerada a partir dos dados obtidos, trata-se da probabilidade posteriori;
- $p(d|h)$: probabilidade dos dados serem observados a partir da hipótese inicial. Essa é a função de verossimilhança e é uma medida de quão bem a função de hipótese se ajusta aos dados;
- $p(h)$: distribuição de probabilidade priori acerca da hipótese;
- $\int_{h'} p(d|h') \times p(h') dh'$: podemos pensar nesse termo como um termo de normalização das probabilidades. Trata-se da verossimilhança marginal.

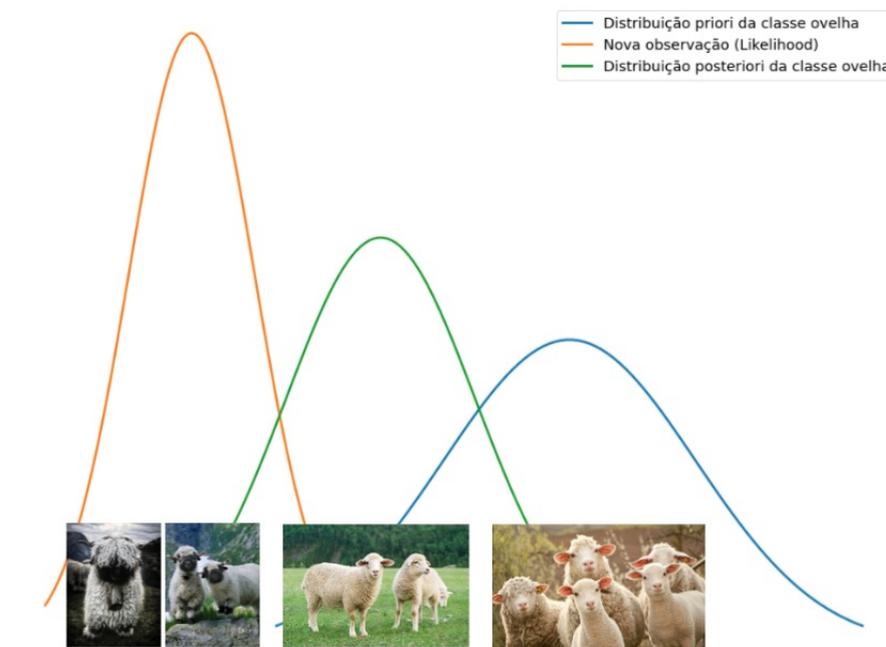
Podemos tomar o seguinte exemplo para elucidar o processo de aprendizagem com o teorema de Bayes: uma criança possui uma certa hipótese acerca das características que permitem classificar um determinado animal como uma ovelha. Essa seria sua distribuição de probabilidade priori, que determina uma probabilidade de um animal ser classificado como uma ovelha dado suas características. Ao encontrar um novo animal e ser indicado que aquele pode ser classificado como uma ovelha, a criança obtém uma distribuição que maximiza a verossimilhança para aquele novo dado, ou seja, uma distribuição que melhor explica aquele novo animal. Ela então atualiza suas crenças, combinando seu conhecimento com a nova função de verossimilhança, obtendo uma distribuição posteriori, que substitui seu conhecimento inicial. Essa ideia é ilustrada nas Figuras 1 e 2.

Figura 1 – Distribuição a priori da classe ovelhas



Fonte: Adaptação, (FREITAS, 2013)

Figura 2 – Distribuição posteriori após o aprendizado



Fonte: Adaptação, (FREITAS, 2013)

2.1.1 Exemplo com Regressão Linear

Nesta seção, utilizaremos o exemplo do método sendo aplicado para a realização de uma regressão linear simples. Seja na utilização de modelos simples como uma regressão

linear ou em modelos tão complexos quanto se queira, o algoritmo da inferência estatística bayesiana se mantém o mesmo:

1. Seja um conjunto de dados $D = \{x_1, x_2, \dots, x_n\}$, determina-se a função de verossimilhança $P(D|\theta)$;
2. Especifica-se uma distribuição de probabilidade priori $P(\theta)$;
3. Computa-se a probabilidade posteriori com o teorema de Bayes: $P(\theta|D) = \frac{P(D|\theta) \times P(\theta)}{P(D)}$.

No caso da regressão linear com um ruído gaussiano, temos:

$$f(x) = x^T \theta, y = f(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_n^2) \quad (2.2)$$

De acordo com (RASMUSSEN; WILLIAMS, 2006), para o problema da regressão linear, a função de verossimilhança é dada por:

$$L(y|X, \theta, \sigma^2) = \mathcal{N}(y|X\theta, \sigma^2 \times I_n)$$

Ao afirmar que a verossimilhança é Gaussiana, estamos afirmando que dado um θ e um σ^2 , a probabilidade desse modelo explicar os dados y_i é modelado por uma Gaussiana, ou seja:

$$L(y|X, \theta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - x\theta)^2}{2\sigma^2}\right)$$

Seguindo o algoritmo proposto, devemos especificar uma distribuição priori para o modelo. Nesse caso, como o modelo é uma simples regressão linear, especificamos uma distribuição priori para o parâmetro θ Gaussiana com média 0 e matriz de variância V_p .

$$\theta \sim \mathcal{N}(0, V_p)$$

Novamente, dado que temos uma distribuição priori e uma função de verossimilhança, podemos calcular a distribuição posteriori utilizando o teorema de bayes (equação 2.1). Pode-se mostrar que, nesse caso, a distribuição posteriori é dada por uma Gaussiana com média $\bar{\theta}$ e matriz de covariância A^{-1} :

$$p(\theta|X, y) \sim \mathcal{N}(\bar{\theta}, A^{-1}) \quad (2.3)$$

em que:

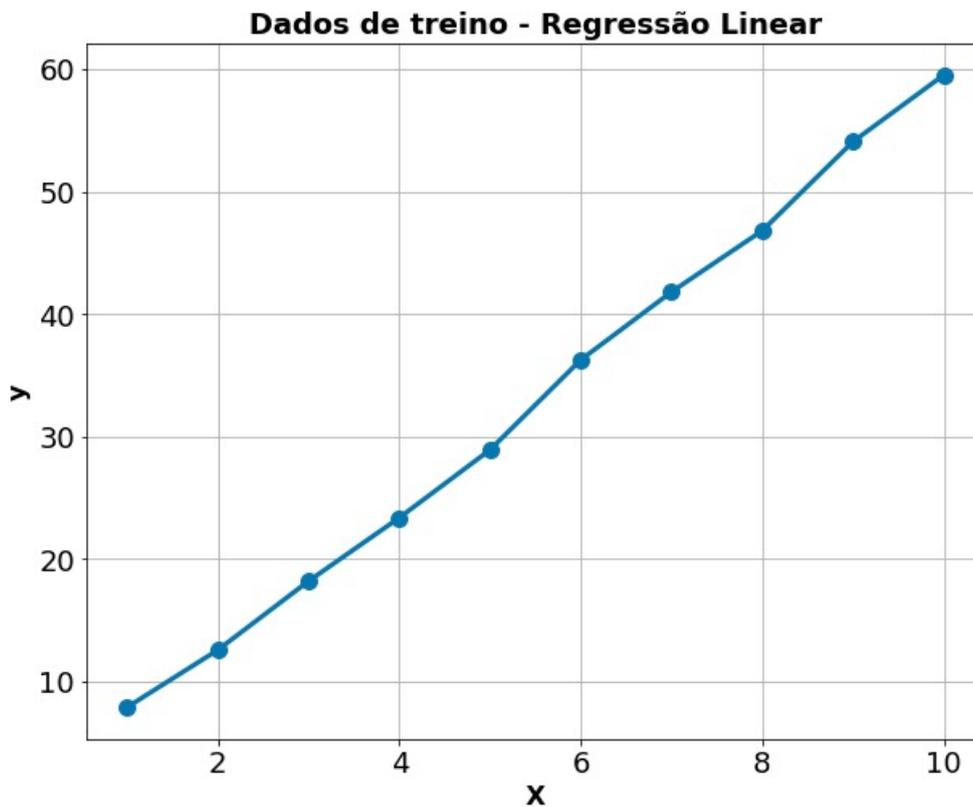
$$A = \sigma_n^{-2} X X^T + V_p^{-1} \quad (2.4)$$

$$\bar{\theta} = \frac{1}{\sigma_n^2} A^{-1} X y \quad (2.5)$$

É importante notar que, como a distribuição priori e a função de verossimilhança são Gaussianas, a distribuição posteriori também será. Podemos perceber que tanto a média da nova distribuição quanto sua matriz de covariância dependem dos dados de treinamento.

Para a visualização do treinamento, definimos um conjunto de dados unidimensional com 10 observações $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. A variável alvo foi criada por um modelo linear com um θ verdadeiro igual a 6 somada a um ruído gaussiano com média 0 e variância igual a 1. A Figura 3 ilustra os dados gerados.

Figura 3 – Dados de treinamento para a regressão linear



Fonte: Autoria Própria, 2021

Destaca-se que é necessário que tenhamos uma definição inicial acerca da distribuição priori dos dados e que essa acaba sendo uma das principais dificuldades da aprendizagem bayesiana, ou seja, a hipótese inicial acerca da função priori é muito importante no processo de aprendizagem. De acordo com a certeza anterior que se possui acerca do fenômeno, é possível gerar uma distribuição priori "forte" ou "fraca". Uma distribuição forte é uma distribuição que possui uma variância muito baixa e a curva Gaussiana se

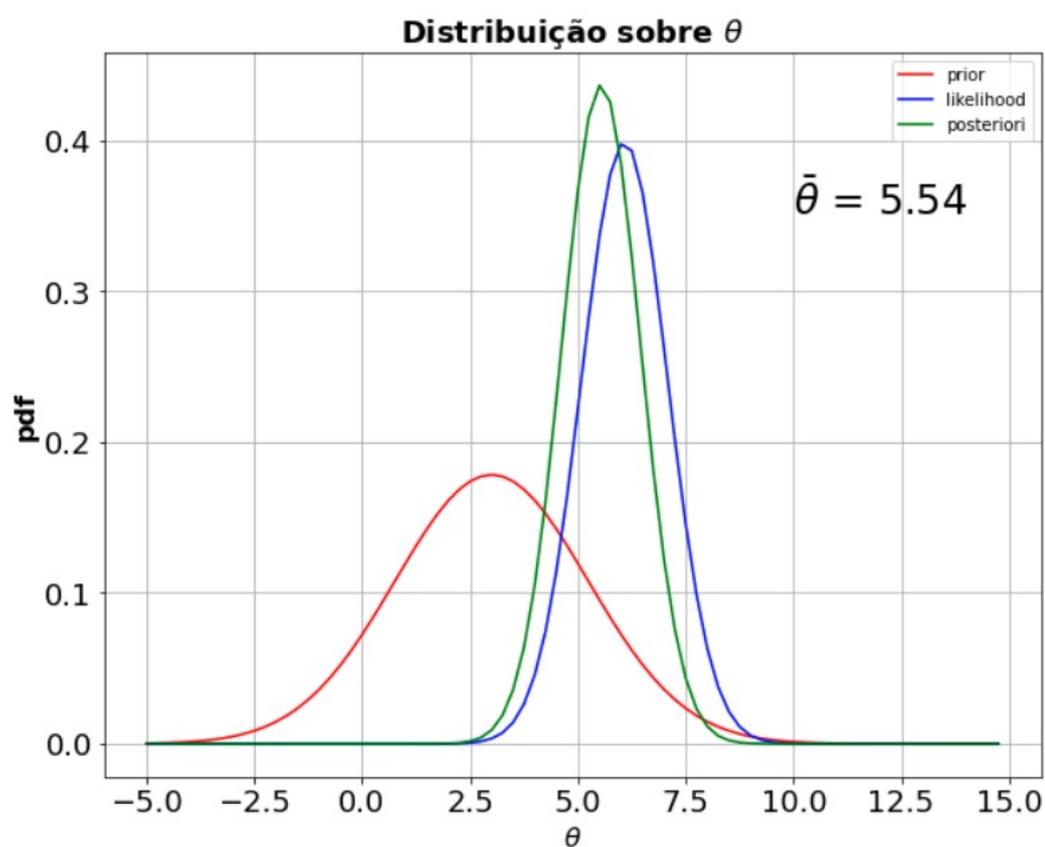
torna bastante concentrada. Uma distribuição "fraca", por sua vez, possui uma variância alta e a distribuição Gaussiana se torna menos concentrada.

Para observar a distribuição posteriori gerada a partir de hipóteses prioris fortes e fracas, realizamos o processo de aprendizagem com duas Gaussianas iniciais:

- $\mathcal{N}(3, 5)$ - Gaussiana com média 3 e variância 5 (fraca)
- $\mathcal{N}(3, 1)$ - Gaussiana com média 3 e variância 1 (forte)

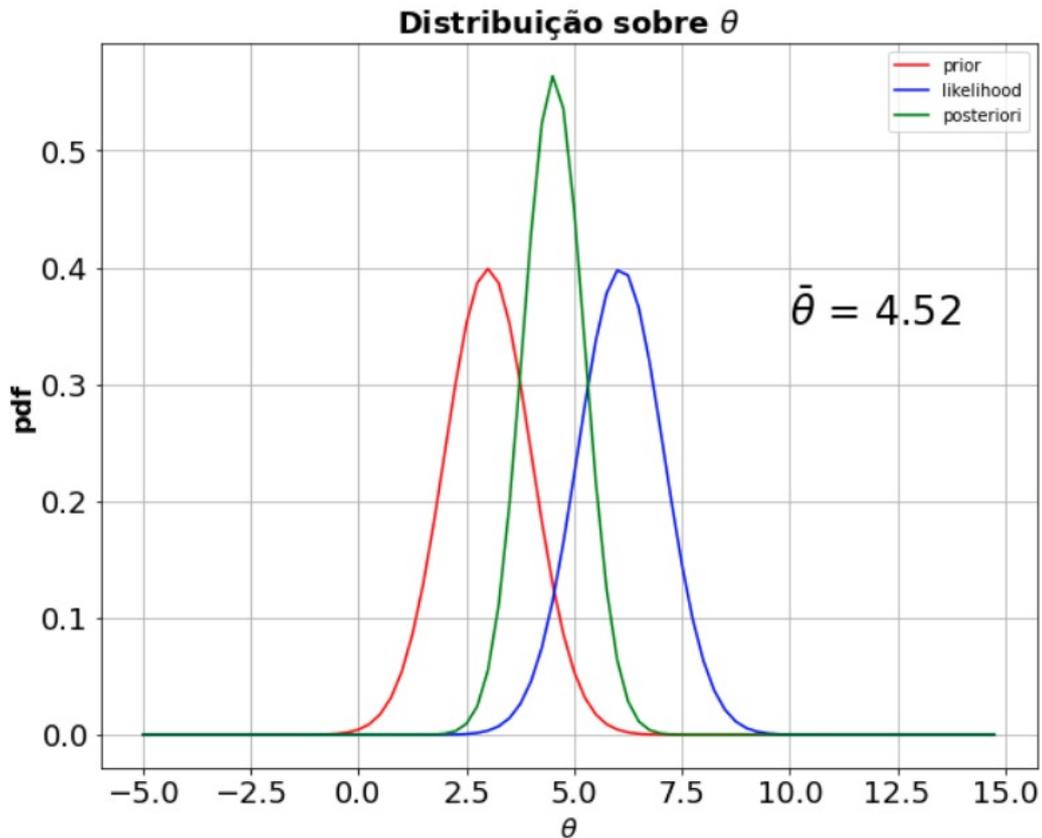
O resultado obtido para o treinamento pode ser observado nas Figuras 4 e 5.

Figura 4 – Inferência utilizando uma distribuição priori fraca



Fonte: Autoria Própria, 2021

Figura 5 – Inferência utilizando uma distribuição priori forte



Fonte: Autoria Própria, 2021

Podemos observar que com uma suposição forte na distribuição inicial, o ajuste se deu de forma inferior, já que o θ obtido na distribuição posteriori ficou mais distante do θ verdadeiro. É importante notar, também, que a função prior possuía um θ médio igual a 1, o que era distante do θ verdadeiro (igual a 6).

2.1.2 Gerando previsões a partir do modelo treinado

Para realizar previsões em um caso de teste, é necessário ponderar sobre toda a distribuição posteriori do parâmetro θ . Ou seja, ao contrário da abordagem frequentista na qual a previsão é dada apenas pelo produto entre o θ aprendido e uma nova entrada, na abordagem bayesiana existe uma ponderação sobre infinitos valores de θ pela curva de densidade Gaussiana posteriori aprendida. Assim, a distribuição preditiva para um $f(x_*)$ é dado pela equação 2.6.

É interessante notar que o termo da variância também é ponderado em relação ao novo dado de entrada. Isso significa que, pela aprendizagem bayesiana, é possível atribuir

uma variância menor em locais em que os dados estão mais concentrados.

$$p(f_*|x_*, X, y) = \int p(f_*|x_*, \theta)p(\theta|X, y)d\theta = \mathfrak{N}\left(\frac{1}{\sigma_n^2}x_*^T A^{-1}Xy, x_*^T A^{-1}x_*\right) \quad (2.6)$$

2.2 Método de Regressão por Processos Gaussianos

Na seção passada, observamos como podemos utilizar a aprendizagem bayesiana para encontrar uma distribuição de probabilidade preditiva sobre um parâmetro θ . Na Regressão por Processo Gaussiano, essa ideia se mantém a mesma, porém, em vez de encontrarmos uma distribuição de probabilidade sobre um único parâmetro, realizaremos um processo de inferência sobre toda uma classe de funções f . Podemos, intuitivamente, pensar que essa classe de função é o "parâmetro" do modelo (SWASTANTO, 2016).

De acordo com (RASMUSSEN; WILLIAMS, 2006), "Um processo gaussiano é uma coleção de variáveis aleatórias, em que qualquer número finito das quais possui uma distribuição conjunta de Gauss".

Podemos afirmar que um processo gaussiano é completamente definido pela sua função de média $m(x)$ e sua função de covariância - $k(x, x')$ (também chamada de *kernel*). Podemos especificá-las de acordo com as equações 2.7 e 2.8.

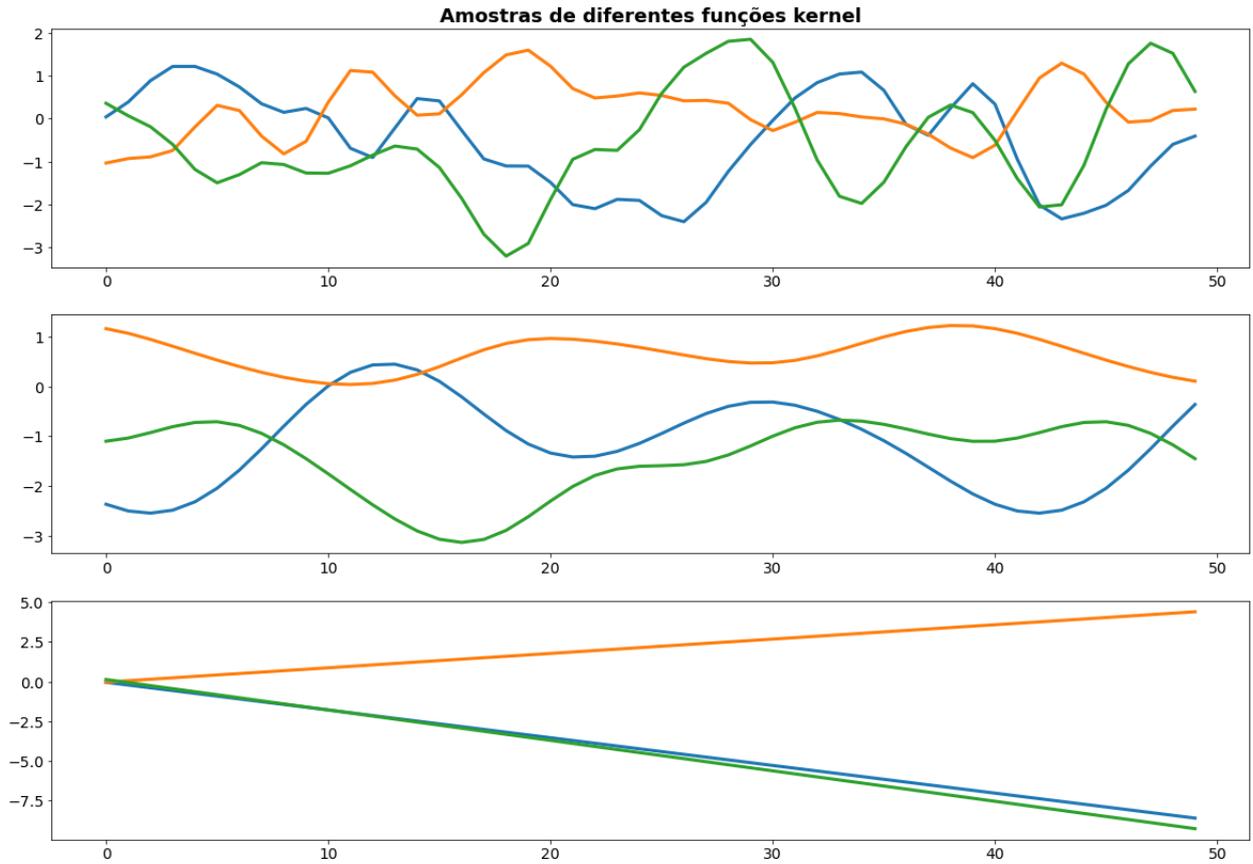
$$m(x) = E[f(x)] \quad (2.7)$$

$$k(x, x') = E[(f(x) - m(x))(f(x') - m(x')))] \quad (2.8)$$

Uma forma comum de representação do processo gaussiano é dado por $f(x) \sim GP(m(x), k(x, x'))$. Por razões práticas e também para evitar a introdução de um viés sobre a tendência do modelo, geralmente se utiliza uma função de média igual a zero (ABDESSALEM et al., 2017).

Depois de levar a média em consideração, o tipo de estrutura que pode ser capturado pelo método GPR é determinado inteiramente pelo seu *kernel*. O *kernel* determina como o modelo generaliza ou extrapola para os novos dados. Alguns exemplos de funções *kernels* podem ser observados na Figura 6. No primeiro gráfico, observamos um *kernel* com características de suavidade. O segundo apresenta um *kernel* com características periódicas e o último, por sua vez, mostra um *kernel* com características lineares. Mais detalhamento acerca dessas funções de covariância e suas características é dado na seção seguinte.

Figura 6 – Exemplos de funções Kernel



Fonte: Adaptação, (SWASTANTO, 2016)

Seguindo a definição, o valor de uma função f avaliada em um ponto x possui uma distribuição Gaussiana multidimensional, ou seja:

$$\begin{bmatrix} f(x_i) \\ f(x_j) \\ f(x_k) \end{bmatrix} = \mathcal{N} \left(\begin{bmatrix} m(x_i) \\ m(x_j) \\ m(x_k) \end{bmatrix}, \begin{bmatrix} k(x_i, x_i) & k(x_i, x_j) & k(x_i, x_k) \\ k(x_j, x_i) & k(x_j, x_j) & k(x_j, x_k) \\ k(x_k, x_i) & k(x_k, x_j) & k(x_k, x_k) \end{bmatrix} \right) \quad (2.9)$$

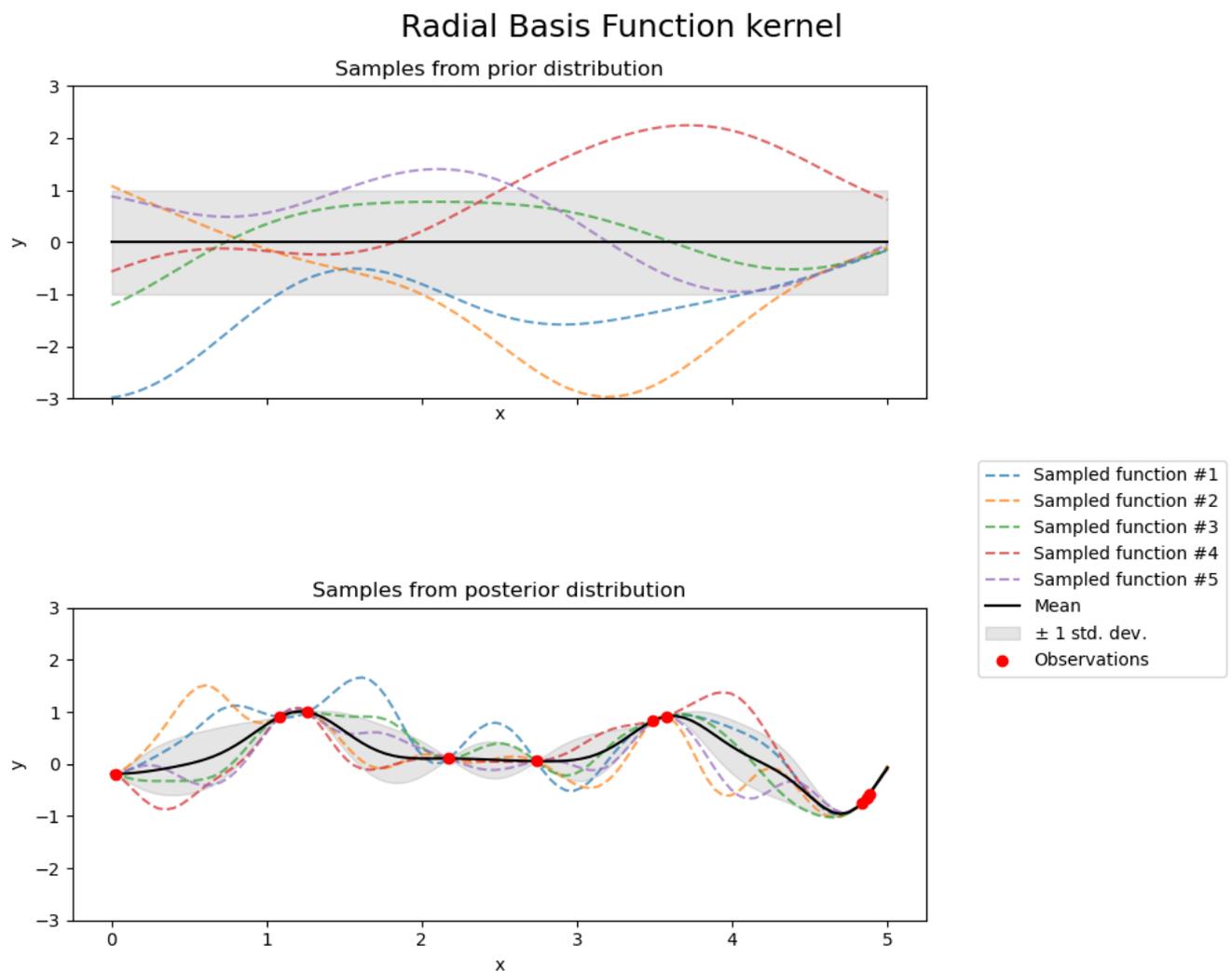
Para calcular a distribuição posteriori, seguindo a mesma abordagem proposta na seção anterior, podemos assumir que temos observações conhecidas (conjunto de treino), dadas por $(x_i, f_i) | i = 1, 2, \dots, n$. A distribuição Gaussiana conjunta das observações de treinamento \mathbf{f} e uma observação de teste, dado a distribuição priori é dada pela equação 2.10 (RASMUSSEN; WILLIAMS, 2006).

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right) \quad (2.10)$$

Se existirem n exemplos de treinamento e n_* exemplos de teste, então $K(X, X_*)$ denota uma matriz $n \times n_*$ de covariâncias avaliada entre todos os pares de treino e de teste. O mesmo vale para as demais matrizes $K(X, X)$, $K(X_*, X)$ e $K(X_*, X_*)$.

Para obter a distribuição posteriori, devemos restringir a distribuição priori para conter apenas as funções que englobam os pontos de dados observados. Na Figura 7, podemos observar esse processo. Inicialmente observamos 5 funções amostras (nas cores vermelha, verde, amarelo, azul e roxa; e em tracejado) de um *kernel* que possui uma característica de suavidade. Em seguida, observamos o processo de ajuste do modelo de regressão Gaussiana de acordo com os dados de treinamento (pontos em vermelho). A linha preta cheia nos indica a média prevista pelo processo gaussiano, ao passo que a região cinza indica a região de ± 1 desvio padrão. Podemos notar que tanto a média do processo quanto seu intervalo de confiança são afetados pela etapa de treinamento.

Figura 7 – Amostras a priori e a posteriori



De maneira formal, podemos afirmar que a equação de predição gerada a partir dos dados é estabelecida pela equação 2.11 (RASMUSSEN; WILLIAMS, 2006).

$$f_*|X_*, X, f \sim \mathfrak{N}(K(X_*, X)K(X, X)^{-1}f, K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)) \quad (2.11)$$

Dessa forma, podemos indicar que o processo de treinamento com o GPR é dado da seguinte maneira:

- Define-se uma distribuição de funções prioris amostradas de uma Gaussiana com média 0 e função de covariância K. As características dessa função devem ser adequadas para cada tipo de problema.

$$p(f|X) \sim \mathfrak{N}(f|m(X), K(X, X))$$

- Obtém-se a função de verossimilhança de maneira similar ao caso da regressão linear da seção anterior. Nesse caso específico, a média da verossimilhança é centrada em uma avaliação de \mathbf{f} arbitrária. A variância é dada por um σ_n^2 previamente assumido.

$$p(y|f) \sim \mathfrak{N}(y|f, \sigma_n^2 I)$$

- Por fim, utiliza-se a atualização por aprendizagem bayesiana para calcular a distribuição posteriori, dada por (SWASTANTO, 2016)

$$p(f|X, y) \sim \mathfrak{N}(f|\bar{\mu}, \bar{\Sigma})$$

, em que:

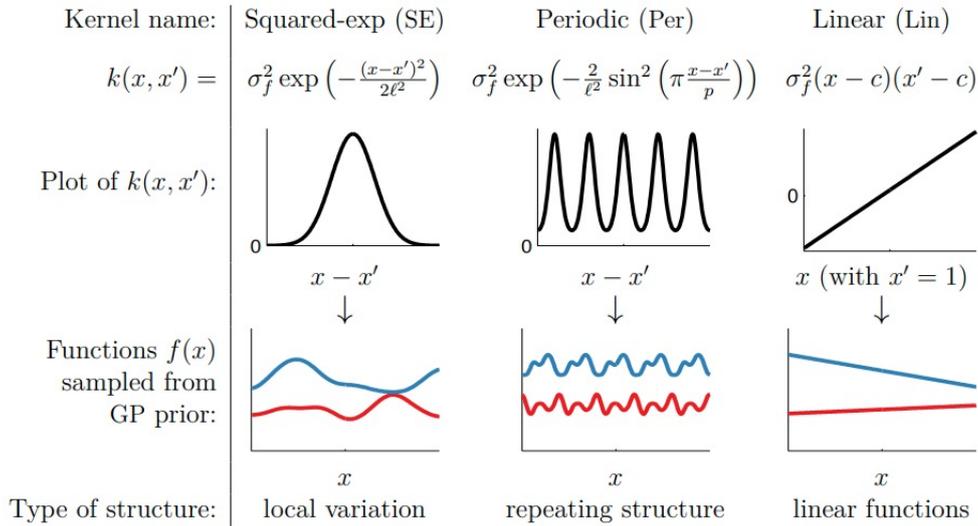
$$\begin{aligned} - \bar{\mu} &= K(X_*, X)k(X, X)^{-1}f \\ - \bar{\Sigma} &= K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*) \end{aligned}$$

2.2.1 Funções Kernel

A maior dificuldade de se realizar uma tarefa de aprendizagem por meio de um GPR é a escolha de uma função de covariância (*kernel*) adequada para o problema (FREITAS, 2013). É necessário que o desenvolvedor do modelo esteja familiarizado com os dados do determinado problema e com as características gerais da função geradora. Em diversos pacotes de implementação do algoritmo GPR no Python, é possível utilizar funções *kernel* definidas pelo usuário.

Dito isso, podemos apresentar algumas funções *kernel* bastante conhecidas na literatura e que são capazes de se combinarem, gerando ainda mais funções *kernel* derivadas. Na Figura 8, são mostrados três *kernels* básicos, que serão detalhados a seguir.

Figura 8 – Funções *kernel* base



Fonte: (DUVENAUD, 2014)

2.2.1.1 Kernel Squared Exponential (SE)

O *kernel squared exponential* (do inglês, exponencial quadrático) também é conhecido como *Radial Basis Function Kernel* (RBF) e possui a forma dada pela equação 2.12.

$$K_{se}(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) \tag{2.12}$$

O *kernel* SE é um *kernel* do tipo estacionário. Isso significa que o valor da sua função K_{se} depende apenas da diferença $x - x'$. Na Figura 8, podemos ver, na segunda, linha o gráfico do valor da função do *kernel* de acordo com o valor da diferença $x - x'$. Na terceira linha podemos ver amostras de Gaussianas construídas por meio de uma função de covariância que segue a regra do *kernel* SE.

Na equação 2.12, o parâmetro l é chamado de *lengthscale* e determina o tamanho das ondulações observadas nas amostras. A variância σ^2 , por sua vez, determina a distância média de um valor da função da sua média.

2.2.1.2 Kernel Periódico

O *kernel* periódico permite que modelemos funções que se repetem ao longo do tempo. A sua utilização é recomendada em problemas de séries temporais que apresentam

sazonalidade. Tal função *kernel* é dada pela equação 2.13.

$$K_{Per}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right) \quad (2.13)$$

Nessa estrutura, o parâmetro p determina a distância entre os ciclos da função e o parâmetro l possui o mesmo significado que no *kernel* anterior.

2.2.1.3 Kernel Linear

O *kernel* linear, ao contrário dos anteriores é um *kernel* não estacionário. Isso significa que o modelo GPR produzirá diferentes previsões caso haja um deslocamento linear nos dados (DUVENAUD, 2014).

Se utilizarmos apenas o *kernel* linear para um determinado problema, estaremos implementando uma simples regressão bayesiana, conforme descrita na seção anterior. O valor da função desse *kernel* é dada pela equação 2.14

$$K_{lin}(x, x') = \sigma(x - c)(x' - c) \quad (2.14)$$

em que o parâmetro c é chamado de *offset* e determina um ponto em que todas as funções amostras da distribuição posteriori passam.

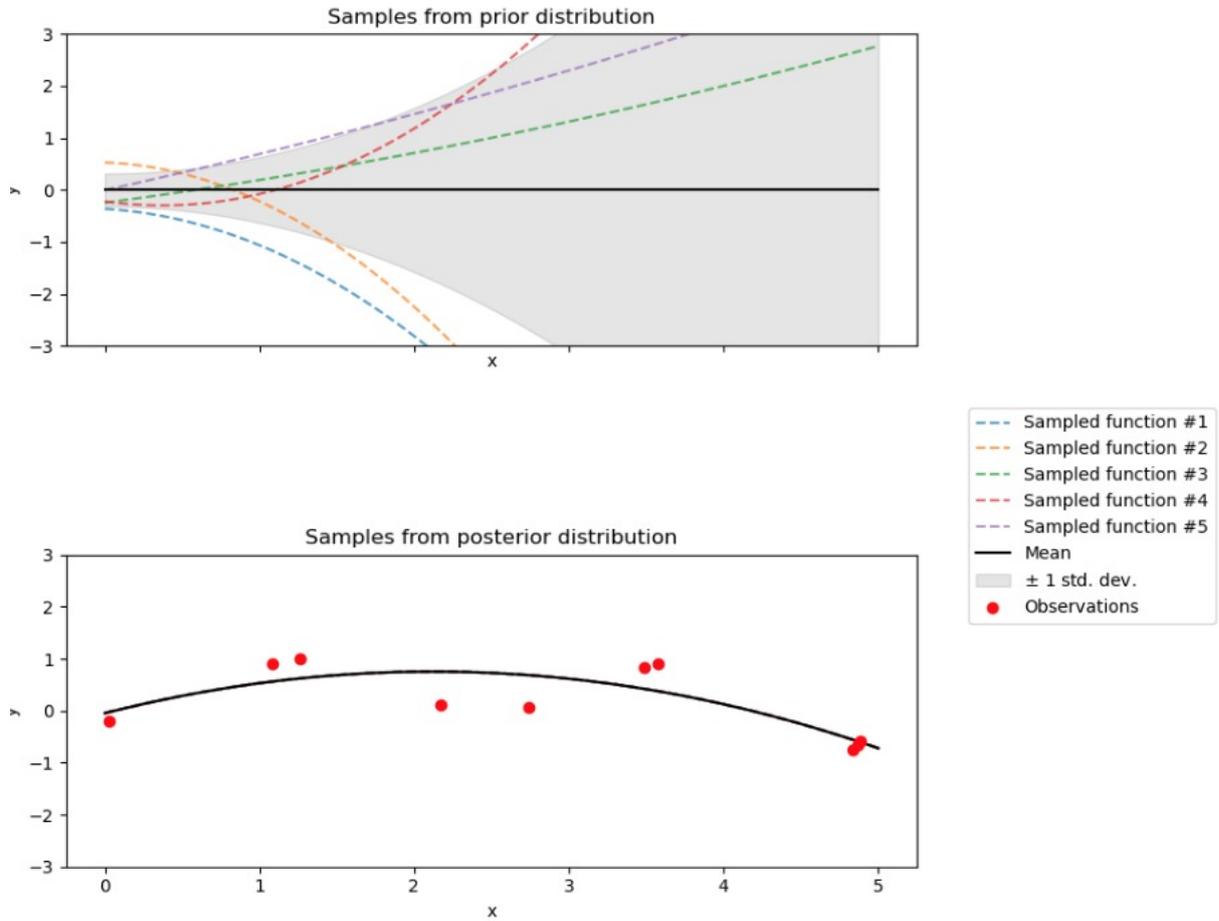
O *kernel* linear pode ser combinado com ele mesmo para construir polinômios de ordens superiores. A multiplicação, por exemplo, de um *kernel* linear com um *kernel* linear produz um *kernel* que modela uma função quadrática. Um exemplo de um *kernel* linear de ordem 2 é mostrado na Figura 9.

2.2.2 Estruturas de Kernels

Além dos *kernels* base e de *kernels* que podem ser definidos pelo usuário, é possível combinar *kernels* por meio de operações de adição e de multiplicação. Qualquer quantidade de *kernels* pode ser combinada entre si, porém, problemas de convergência podem ocorrer no treinamento do modelo GPR, já que nem todas as combinações vão gerar uma matriz de covariância positiva semidefinida.

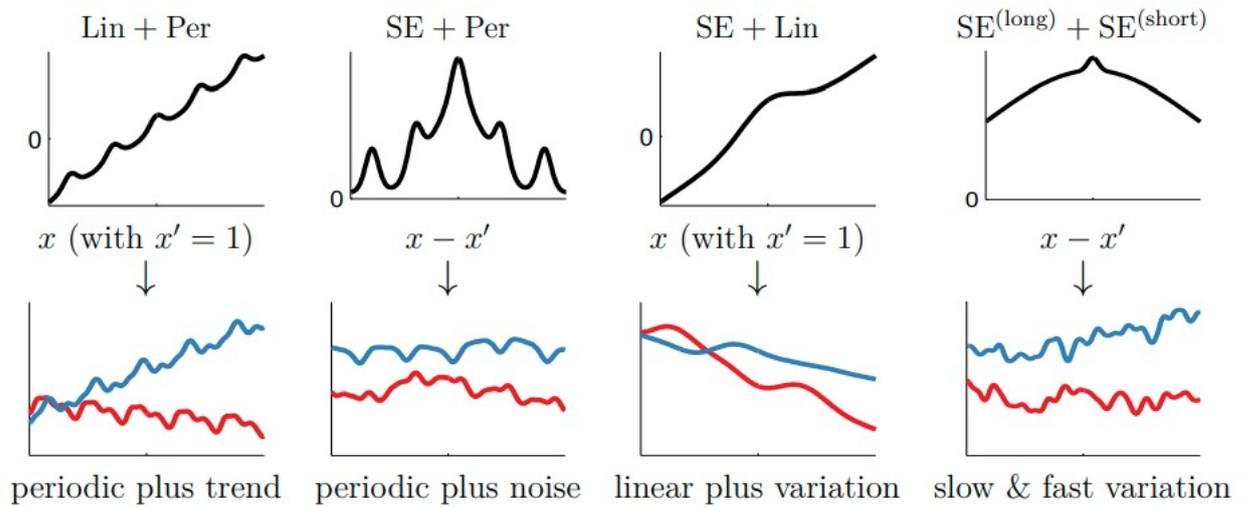
Nas Figuras 10 e 11, podemos ver exemplos de novas funções *kernels* geradas a partir de combinações das funções base. O estudo da decomposição das séries temporais pode nos indicar que *kernels* devemos utilizar para determinado tipo de problema.

Figura 9 – Exemplo de treinamento com função *kernel* linear de expoente 2

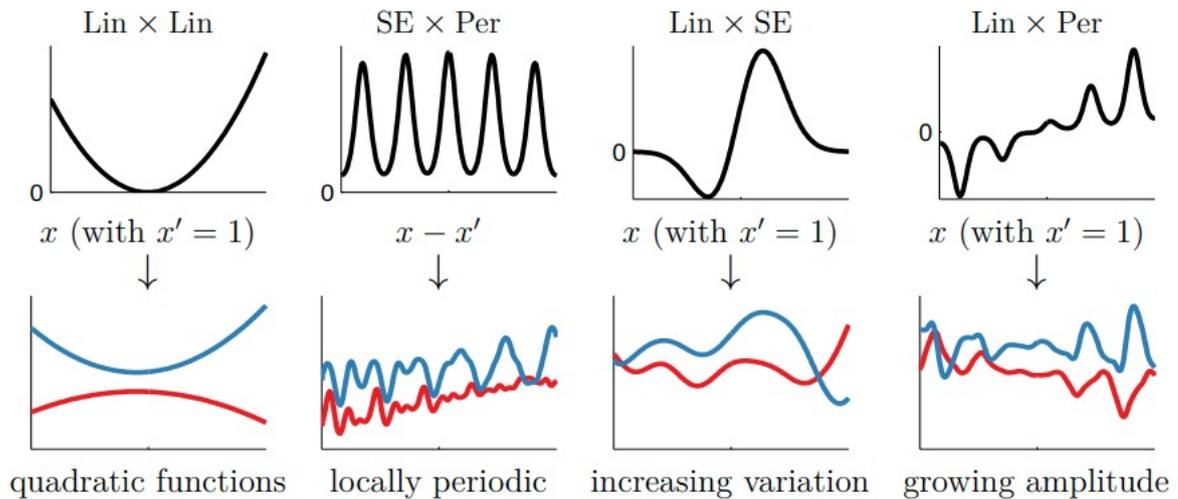


Fonte: (JAN; GUILLAUME, 2021)

Figura 10 – Combinações de funções *kernel* base por meio de soma



Fonte: (DUVENAUD, 2014)

Figura 11 – Combinações de funções *kernel* base por meio de multiplicação

Fonte: (DUVENAUD, 2014)

Na Figura 10, podemos ver 4 exemplos de combinações de kernels:

- Linear + Periódico: esse *kernel* é capaz de capturar uma série que possui tendência e sazonalidade aditiva;
- SE + Periódico: o *kernel* SE é capaz de capturar variações suaves ao longo dos dados. Ao somá-lo com um *kernel* periódico, obteremos uma periodicidade com variações aleatórias;
- SE + Linear: nesse caso, obteremos uma tendência advinda do *kernel* linear somado com variações suaves do *kernel* SE;
- SE(long) + SE(short): essa estrutura também é conhecida como o *kernel* racional quadrático e é dado pela soma de vários *kernels* SE com diferentes valores de *lengthscale*. Aqui esperamos observar funções que variam de maneira suave ao longo de varios *lengthscales*.

Na Figura 11, também observamos 4 exeplos de combinações multiplicativas de kernels:

- Linear x Linear: conforme mencionado, essa estrutura é capaz de gerar polinômios de graus superiores. Uma multiplicação de n *kernels* lineares gera um polinômio de grau n .

- SE x Periódico: essa é uma combinação particularmente importante, já que é capaz de capturar periodicidades locais. O resultado desse *kernel* são funções que são periódicas, mas que variam lentamente ao longo do tempo. Isso adiciona flexibilidade ao modelo.
- Linear x SE: a multiplicação desses *kernels* gera uma estrutura de variações que aumentam em amplitude ao longo do tempo
- Linear x Periódico: por fim, essa configuração também é de grande importância, já que é capaz de modelar séries temporais com tendência e sazonalidade multiplicativa.

De acordo com (DUVENAUD, 2014), as seguintes estruturas de *kernels* podem ser utilizadas para os seguintes problemas de regressão, conforme mostra a Tabela 1.

Tabela 1 – Estruturas de Kernels

Modelo de Regressão	Estrutura do Kernel
Regressão Linear	Constante + Linear + Ruído
Regressão Polinomial	Constante + \prod Linear + Ruído
Regressão semi-paramétrica	Linear + SE + Ruído
Decomposição de Fourier	Constante + \sum Periódico + Ruído
Tendência, sazonalidade e irregularidades	\sum SE + \sum Periódico + Ruído

Fonte: (DUVENAUD, 2014)

2.3 Séries Temporais

Dentro do escopo deste trabalho, estamos utilizando o método de regressão por processos gaussianos para a previsão de casos futuros de Covid-19 na cidade de Campina Grande - PB. Embora essa seja uma tarefa tratada como aprendizagem supervisionada, é importante destacar que trata-se de um conjunto de dados que constituem uma série temporal. Assim, torna-se importante apresentar as principais características deste tipo de série, bem como o seu modo de utilização em problemas de Aprendizagem de Máquina.

Uma série temporal é uma sequência de observações tomadas sequencialmente no tempo (BOX et al., 2016). Diversos fenômenos constituem séries temporais, tais como quantidade de vendas em uma loja por mês, a evolução do preço de ações no mercado financeiro, a temperatura diária de uma determinada cidade ou a quantidade de pacientes infectados por uma doença.

Diferentemente da aprendizagem supervisionada tradicional, em que a ordem das observações geralmente não afeta o processo de aprendizagem, nas séries temporais um cuidado a mais deve ser tomado. Esse tipo de série adiciona uma ordem explícita de

dependência entre observações que é justamente a dimensão temporal (BROWLEE, 2020). Essa dimensão adicional é tanto uma limitação para o problema quanto uma estrutura que provê uma fonte de informação adicional.

O estudo das séries temporais é geralmente dividido entre a análise e a predição (ou do inglês *time series forecasting*). Embora algumas técnicas de aprendizagem não necessariamente necessitem de uma análise detalhada da série que se propõe a prever, descrições estatísticas são importantes tanto para entender o dado de treinamento, quanto para tomar decisões acerca da maioria dos modelos de previsão.

Uma das principais técnicas para análise da série temporal é a decomposição da série. Uma maneira comum de decomposição é em uma parcela sistemática e em uma parcela não-sistemática (GALIT; KENNETH, 2016). A parcela sistemática é subdividida em nível, tendência e sazonalidade. A parcela não-sistemática, por sua vez, é chamada de ruído. O nível descreve o valor médio da série, a tendência é a variação da série entre períodos e a sazonalidade descreve um comportamento cíclico de curta duração que pode ser observado diversas vezes ao longo da série. Por fim, o ruído é a variação aleatória advinda de erros de medição ou outras razões diversas.

Os componentes da série podem ser aditivos ou multiplicativos. As equações 2.15 e 2.16 mostram a decomposição do sinal da série dessas duas maneiras. A Figura 12 ilustra alguns exemplos de séries temporais com diferentes componentes.

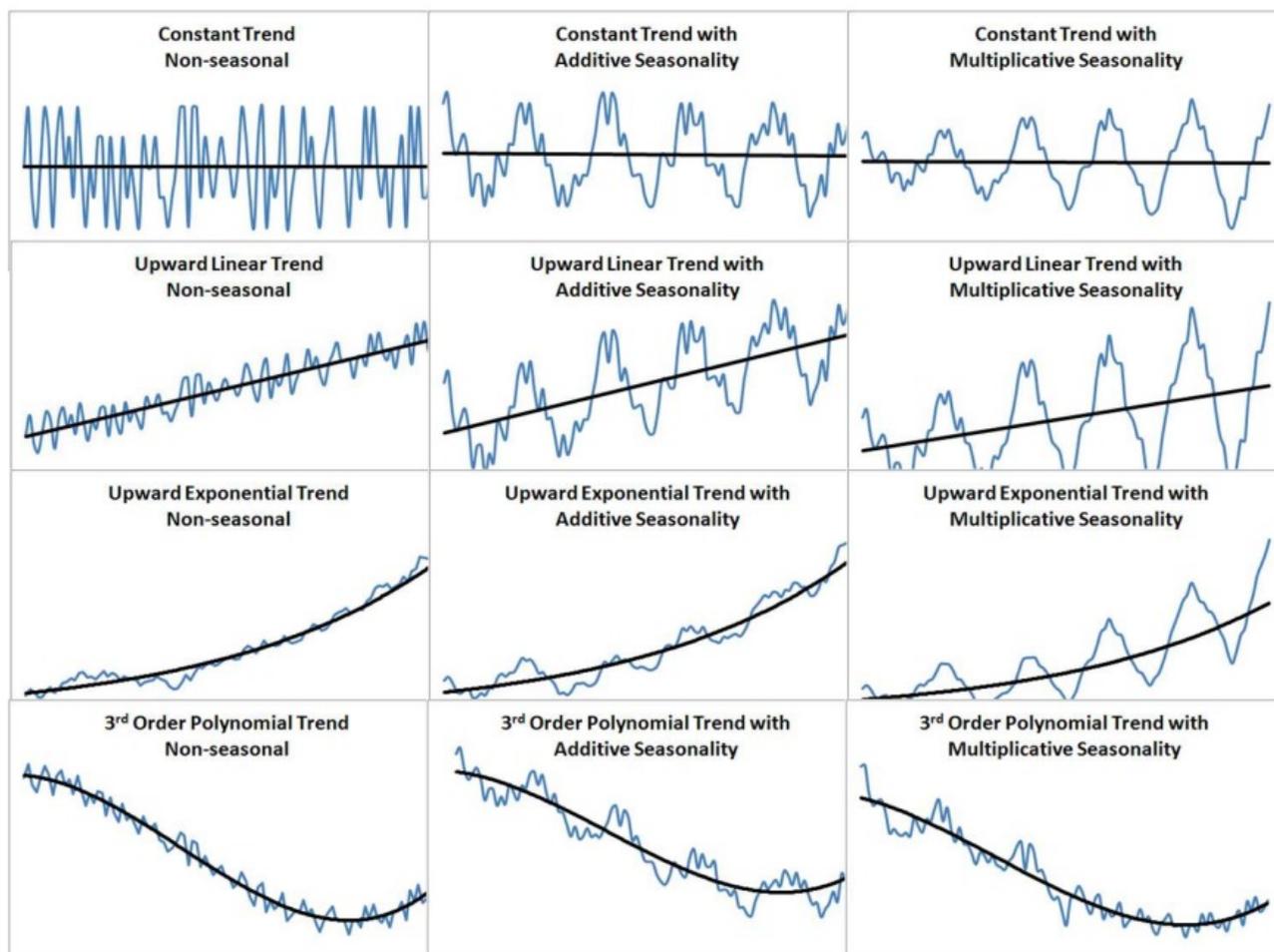
$$y = \text{nível} + \text{tendência} + \text{sazonalidade} + \text{ruído} \quad (2.15)$$

$$y = \text{nível} \times \text{tendência} \times \text{sazonalidade} \times \text{ruído} \quad (2.16)$$

Além da visualização inicial da série temporal e da decomposição em nível, tendência e sazonalidade, também podemos explorar características do sinal por meio do cálculo da autocorrelação. Em uma série temporal, é comum que haja uma correlação entre valores adjacentes, ou seja, um valor passado afeta um valor futuro. O cálculo desse impacto, ou autocorrelação pode ajudar a melhor desenvolver modelos de previsão (GALIT; KENNETH, 2016). A autocorrelação parcial é outra maneira de analisar o impacto dos valores passados nos valores futuros, porém, eliminando a influência de valores intermediários. Por exemplo, dada uma série temporal z_t , a autocorrelação parcial no *lag* k , denotada por $\alpha(k)$ é a autocorrelação entre z_t e z_{t+k} removendo as dependências lineares entre $z_{t+1}, \dots, z_{t+k-1}$.

Dentre principais variáveis regressoras utilizadas em problemas de aprendizagem de máquina, podemos citar as variáveis de atraso (ou do inglês *lag features*) e agregações em janelas de tempo (ou do inglês *window features*). Como exemplo de uma *window feature* podemos citar a média móvel das observações da série ao longo do tempo. Uma das maneiras de decidir o tamanho ou a quantidade dessas variáveis regressoras é observando como as

Figura 12 – Exemplos de séries temporais com diferentes componentes



Fonte: (GALIT; KENNETH, 2016)

observações do passado afetam o futuro, ou seja, utilizando o cálculo da autocorrelação e da autocorrelação parcial.

2.3.1 Métricas de avaliação de previsões

No desenvolvimento de modelos de Aprendizagem de Máquina é importante realizar uma separação entre um conjunto de dados de treino e um conjunto de dados de teste. A avaliação da performance do modelo deve ser feita sobre o conjunto de dados não utilizado no processo de treinamento, ou seja, no conjunto de teste. Isso é feito para a retirada do viés do modelo durante o processo de aprendizagem.

Uma das principais técnicas para realizar isso em um problema tradicional de aprendizagem supervisionada é a separação aleatória dos dados em uma proporção de 80%-20%. Essa abordagem não pode ser utilizada na previsão de séries temporais, já que existe uma dependência temporal entre dados passados e dados futuros. Não se pode

utilizar informações do futuro no treinamento do modelo, já que isso não condiz com a realidade de utilização do estimador.

O elemento mais importante do fornecimento de uma previsão é garantir que essa previsão está sendo gerada somente com os dados que se pode obter no período de tempo presente. Isso significa que deve-se levar em consideração não apenas o momento em que o evento aconteceu, mas também quando o dado ficou disponível para o uso (AILEEN, 2020).

Para a avaliação da estratégia de modelo proposta, pode-se utilizar uma estrutura de *backtesting* conhecida como validação cruzada para séries temporais, ou *time series split cross validation*. A ideia é dividir o conjunto de dados em *splits* sequenciais e realizar o processo de treino e teste de maneira incremental. Por exemplo:

- split 1: treinar com 30% dos dados e testar nos próximos 10%, totalizando os 40% iniciais do conjunto;
- split 2: treinar com 50% dos dados e testar nos próximos 10%, totalizando os 60% iniciais do conjunto;
- split 3: treinar com 70% dos dados e testar nos próximos 10%, totalizando os 80% iniciais do conjunto;
- split 4: treinar com 90% dos dados e testar nos próximos 10%, totalizando os 100% conjunto.

A Figura 13 ilustra esse processo. As bolas verdes constituem o conjunto de treinamento e a bola amarela o conjunto de teste.

No processo de aprendizagem utilizando a validação cruzada, podemos utilizar algumas das principais métricas para a avaliação do modelo, tais como:

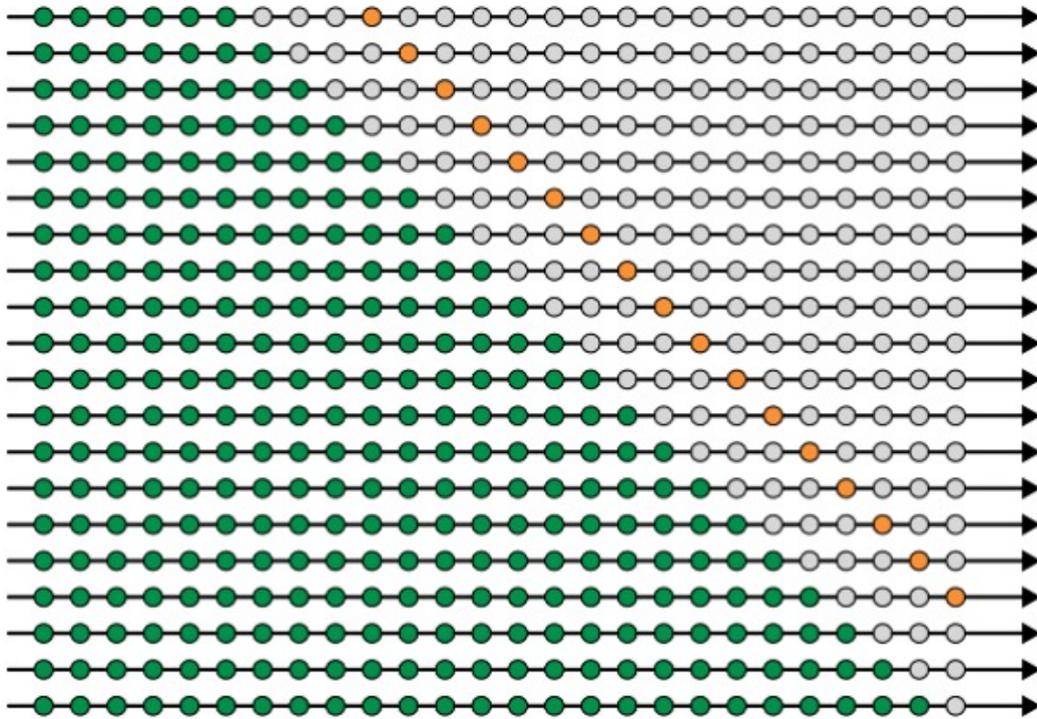
- r^2 : é uma medida de explicabilidade do modelo. Intuitivamente podemos pensar, por exemplo, que se o r^2 possui um valor de 0.8 então o modelo é capaz de explicar 80% da variação dos dados. O r^2 representa a proporção da variância da variável dependente que é explicada pelo modelo de regressão. O valor máximo que pode assumir é igual a 1 e o seu cálculo é dado por:

$$r^2 = 1 - \frac{\sum(f(x_i) - \hat{y})^2}{\sum(y_i - \hat{y})^2} \quad (2.17)$$

- MSE : representa a média da diferença quadrática entre a predição e o dado original. Na prática essa métrica penaliza grandes erros de predição. O MSE é dado por:

$$MSE = \frac{1}{N} \sum (y_i - \hat{y})^2 \quad (2.18)$$

Figura 13 – Ilustração da validação cruzada para séries temporais



Fonte: (AILEEN, 2020)

- *MAPE*: representa o erro percentual médio de predição. É dado por:

$$MAPE = \frac{100\%}{n} \sum \left| \frac{y_i - \hat{y}}{y_i} \right| \quad (2.19)$$

Após obtermos um modelo, devemos observar métricas adicionais para decidir se obtivemos um ajuste adequado aos dados. De acordo com (AILEEN, 2020), dentre as principais técnicas, podemos utilizar:

- **Gráfico das predições do modelo em comparação com o conjunto de teste:** desde que as premissas de geração do conjunto de teste se mantenham similares à do conjunto de treino, devemos observar distribuições de previsão similares ao dado real.
- **Gráfico dos resíduos do modelo:** devemos observar um resíduo homogêneo e com uma distribuição normal

3 Metodologia

De modo a analisar o funcionamento do processo de previsão com o GPR de acordo com a utilização de diferentes funções kernel, três problemas iniciais são propostos:

- previsão de série temporal apenas com componente sazonal
- previsão de série temporal apenas com componente de tendência
- previsão de série temporal com componente de tendência e sazonalidade

Iremos aplicar as métricas de treinamento e avaliação de séries temporais apresentadas na seção 2.3 para observar o comportamento das distribuições prioris e posterioris com o uso de diferentes *kernels* base propostos nas seções 2.2.1.1, 2.2.1.2 e 2.2.1.3.

Após esse estudo inicial, o trabalho segue com a uma aplicação prática do modelo GPR para a previsão de casos acumulados de Covid-19 na cidade de Campina Grande - PB e para a previsão da média móvel de casos na mesma cidade. O principal ferramental necessário para gerar as imagens e para a realização do processo de treinamento foi o Python utilizando os *frameworks pandas, numpy, matplotlib* e o *scikit-learn*. Todo o código pode ser observado no repositório do *github* <https://github.com/andreigor/TCC-GPR>.

3.1 Série temporal apenas com componente de sazonalidade

3.1.1 Dados utilizados

Os dados gerados para este experimento foram adaptados de (ORDUZ, 2019). Primeiramente definimos uma variável temporal com 100 amostras, ou seja, $t = \{1, 2, 3, \dots, 100\}$. Em seguida, definimos uma variável sazonal dada pela equação 3.1, que representa uma senóide.

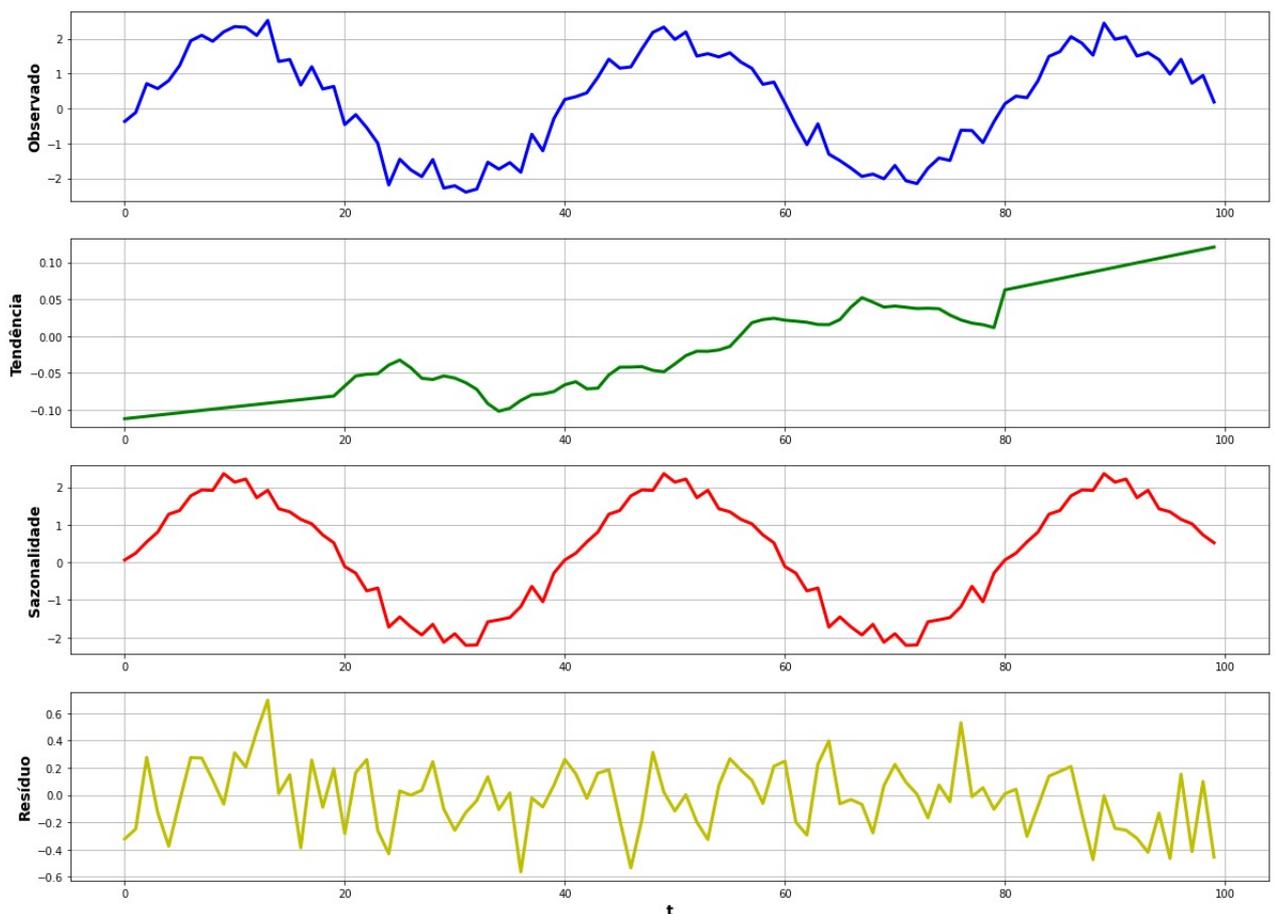
$$y = amplitude \times sen\left(2 \times \pi \times \frac{t}{p}\right) \quad (3.1)$$

A amplitude escolhida para o problema foi de 2, enquanto o período foi definido como 40. A esse sinal y , foi somado um ruído gaussiano (ϵ) com média 0 e variância 0.3, de modo que o sinal alvo proposto para esse problema é dado pela equação 3.2.

$$y_{sazonal} = y + \epsilon \quad (3.2)$$

O sinal $y_{sazonal}$ pode ser observado na Figura 14. Nessa Figura, vemos, em azul o sinal $y_{sazonal}$; em verde vemos a decomposição aditiva (equação 2.15) da tendência do sinal $y_{sazonal}$; em vermelho temos a componente sazonal do sinal observado e, por fim, em amarelo o componente residual (ou de ruído). Podemos perceber que a componente de tendência possui uma ordem de grandeza bem menor que a componente de sazonalidade, de modo que podemos afirmar que a variável alvo é praticamente composta apenas por uma componente sazonal.

Figura 14 – Decomposição aditiva da série temporal do sinal $y_{sazonal}$



Fonte: Autoria própria, 2021

3.1.2 Kernels propostos

Para esse problema, três estruturas de *kernels* foram propostas:

- *kernel 1* - periódico: espera-se que a utilização de um *kernel* periódico realize um ajuste adequado aos dados, dado a natureza sazonal da série temporal
- *kernel 2* - SE: o *kernel* SE é capaz de capturar variações suaves na função alvo, ou seja, espera-se que valores próximos entre as variáveis regressoras devem produzir poucas

variações no resultado da função, enquanto grandes distâncias entre as variáveis regressoras devem produzir grandes variações no resultado da função. Embora essa estrutura seja capaz de realizar ajuste em dados periódicos, esperamos observar um resultado inferior em relação ao *kernel* 1.

- *kernel* 3 - linear: espera-se que a utilização do *kernel* linear não seja adequada para a utilização desse problema, obtendo um resultado ruim nas métricas de avaliação.

O resultado da distribuição das funções priori e posteriori para cada *kernel* proposto, bem como o resultado das métricas r^2 , MSE e MAPE podem ser verificados na seção 4 deste documento.

3.2 Série temporal apenas com componente de tendência

3.2.1 Dados Utilizados

No segundo problema, uma metodologia similar de geração de dados foi utilizada: novamente definimos uma variável temporal com 100 amostras e uma variável de tendência com um grau exponencial de 1.5, mostrado na equação 3.3.

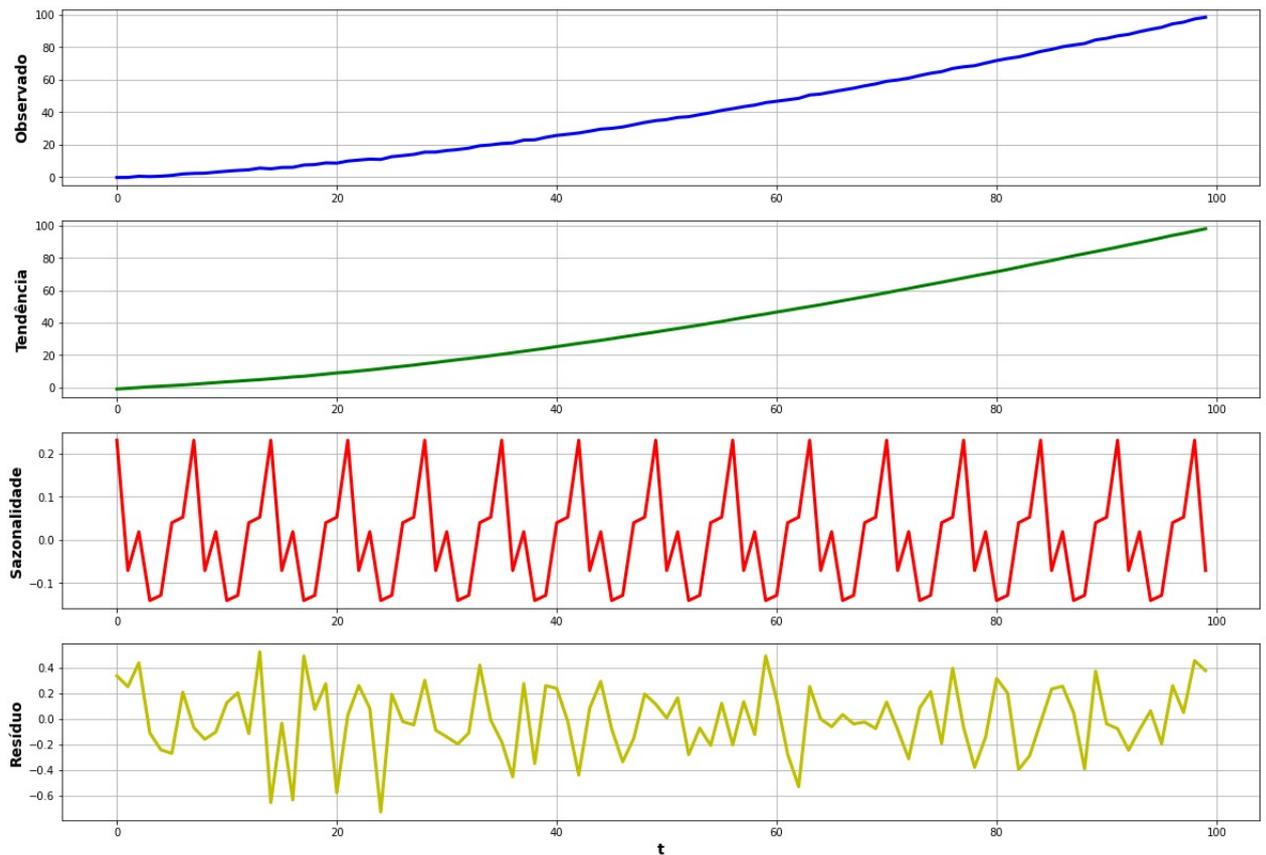
$$y_{tendencia} = t^{1.5} + \epsilon \quad (3.3)$$

O mesmo ruído gaussiano foi somado ao sinal de tendência. O sinal $y_{tendencia}$ pode ser observado na Figura 15. Na Figura, podemos observar o sinal original em azul; em verde temos a parcela de tendência da decomposição aditiva; em vermelho temos a parcela sazonal da decomposição aditiva e, por fim, em amarelo temos o resíduo da decomposição. Nesse caso, podemos perceber que a parcela sazonal dessa série possui uma ordem de grandeza bastante inferior à ordem de grandeza da parcela de tendência.

3.2.2 *kernels* propostos

Para esse problema, as mesmas três estruturas de *kernels* foram propostas:

- *kernel* 1 - periódico: espera-se que a utilização de um *kernel* periódico não seja capaz de realizar um ajuste adequado aos dados, obtendo métricas de avaliações ruins;
- *kernel* 2 - SE: dada a natureza de suavidade do sinal polinomial, pode-se esperar que o *kernel* SE seja capaz de realizar um ajuste aos dados de treino. Porém, esperamos um resultado inferior ao *kernel* linear com um expoente.
- *kernel* 3 - linear: espera-se que a utilização do *kernel* linear seja a mais adequada para esse problema, dada a característica de tendência. Para esse *kernel* específico,

Figura 15 – Decomposição aditiva da série temporal do sinal $y_{tendencia}$ 

Fonte: Autoria própria, 2021

utilizamos o valor da exponenciação igual à 1.5, que é o mesmo da função geradora. Em problemas práticos, não temos acesso à esse tipo de dado, logo, esse é um problema que apresenta um viés e serve apenas como visualização da função *kernel* e do processo de treinamento do GPR.

O resultado da distribuição das funções priori e posteriori para cada *kernel* proposto, bem como o resultado das métricas r^2 , MSE e MAPE podem ser verificados na seção 4 deste documento.

3.3 Série temporal com componente de tendência e de sazonalidade

3.3.1 Dados Utilizados

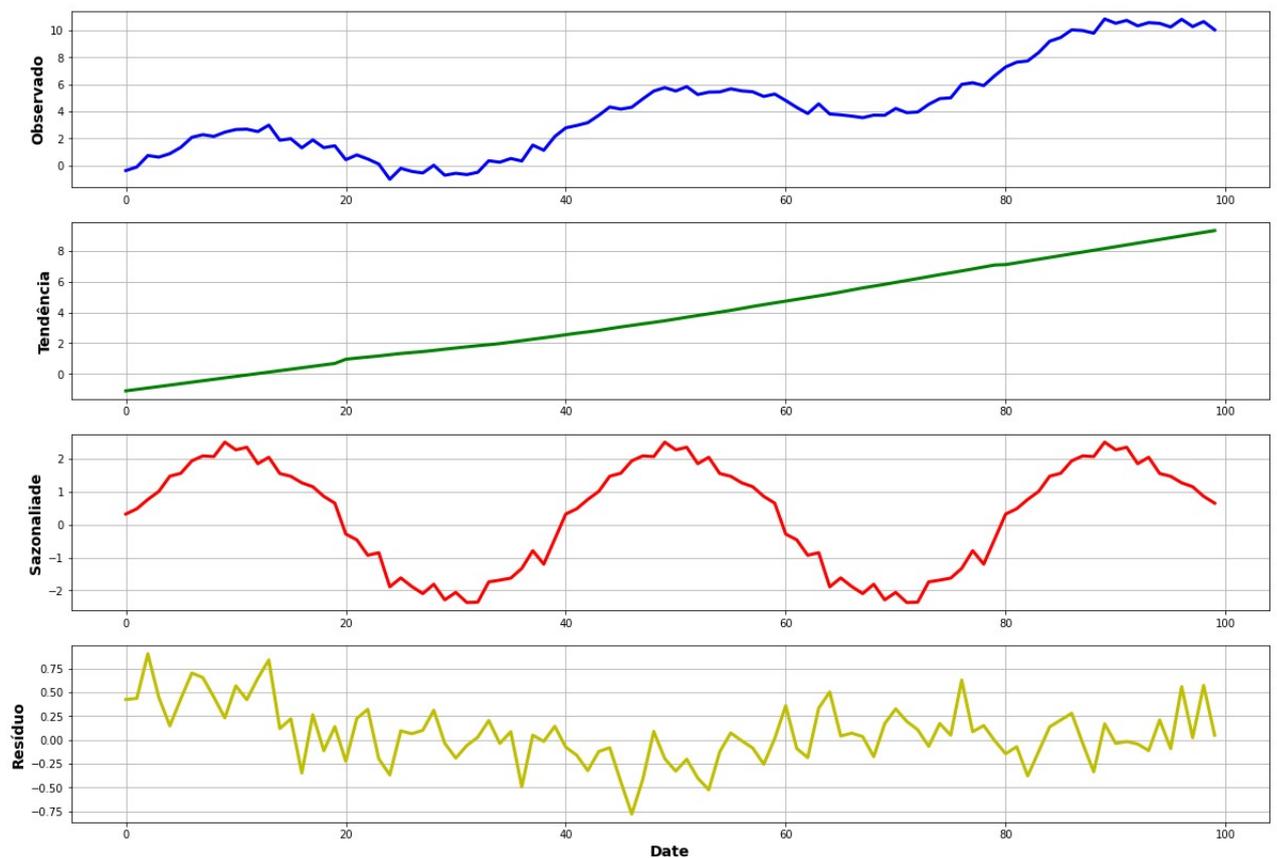
Novamente, para o terceiro problema base, definimos a mesma variável temporal t dos problemas anteriores. Dessa vez, a variável alvo é dada pela soma do sinal de tendência

do problema da seção 3.2 e do sinal sazonal da seção 3.1. A equação que descreve esse sinal é dado a seguir.

$$y_{saz+tend} = amplitude \times sen\left(2 \times \pi \times \frac{t}{p}\right) + t^{1.5} + \epsilon \quad (3.4)$$

O sinal $y_{saz+tend}$ pode ser observado na Figura 16, assim como a sua decomposição aditiva. Diferentemente dos outros problemas, aqui podemos observar uma parcela significativa tanto de tendência quanto de sazonalidade. Dessa maneira, podemos esperar que os *kernels* base não funcionem de maneira adequada para a realização do ajuste aos dados de treinamento. A hipótese a ser testada é a de que a combinação por meio de soma do *kernel* periódico com o *kernel* linear seja a melhor escolha.

Figura 16 – Decomposição aditiva da série temporal do sinal $y_{saz+tend}$



Fonte: Autoria própria, 2021

3.3.2 *Kernels* Propostos

Para esse terceiro problema, além das estruturas de *kernel* base já discutidas, uma quarta estrutura é proposta:

- *kernel* 4 - Linear + Periódico: essa estrutura, conforme discutida na seção 2.2.2 é adequada para a realização de ajustes em séries temporais com componente de tendência e sazonalidade aditiva.

3.4 Previsão do acumulado de casos de Covid

Após a realização do estudo do funcionamento dos *kernels* para os problemas base, partimos para a utilização prática da modelagem com o GPR para a previsão do acumulado de casos de Covid na cidade de Campina Grande - PB. Os dados utilizados foram disponibilizados pela secretaria de saúde da cidade e precisaram passar por uma série de transformações para a sua utilização pelo estimador de Aprendizagem de Máquina.

Na Figura 17, podemos ver as 5 primeiras entradas dos dados brutos disponibilizados. Para fins de visualização, a tabela foi transposta, ou seja, as colunas (ou campos) são mostrados nas linhas, enquanto as entradas de dados são mostradas nas colunas. Isso geralmente é utilizado para visualizar tabelas que possuem muitas colunas, como é o caso aqui.

Figura 17 – Dados brutos de Covid em Campina Grande

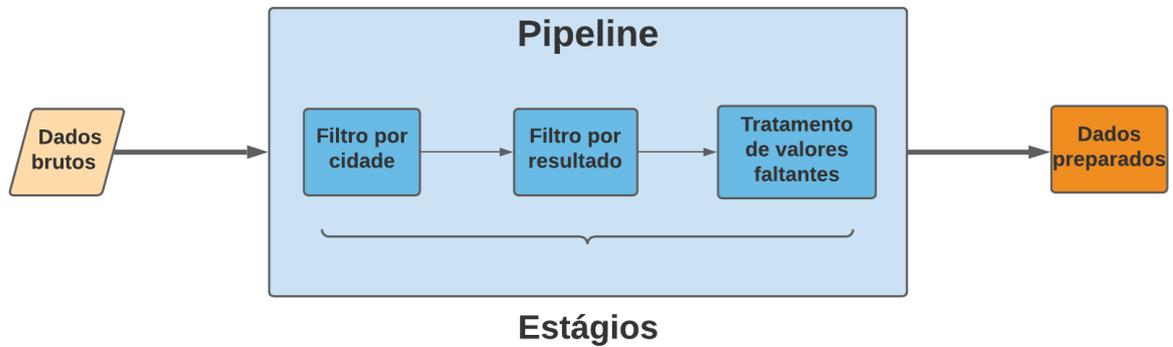
	0	1	2	3	4
Estado	Paraíba	Paraíba	Paraíba	Paraíba	Paraíba
Cidade	Campina Grande	Campina Grande	Campina Grande	Campina Grande	Campina Grande
Teste	TESTE RÁPIDO - ANTICORPO	TESTE RÁPIDO - ANTÍGENO			
Resultado	Negativo	Positivo	Negativo	Positivo	Negativo
Evolucao	NaN	Cura	NaN	NaN	NaN
Data PCR	28/6/2021	NaN	26/6/2021	28/6/2021	28/6/2021
Conclusao	Concluido	Concluido	Concluido	Concluido	Concluido
Data Encerramento	NaN	30/6/2021	NaN	NaN	NaN
Classificacao Final	NaN	Confirmado Laboratorial	NaN	NaN	NaN
Teste Sorologico	NaN	NaN	NaN	NaN	NaN
Data Sorologico	NaN	NaN	NaN	NaN	NaN
Estado Residencia	Paraíba	Paraíba	Paraíba	Paraíba	Paraíba
Sexo	Feminino	Feminino	Feminino	Feminino	Masculino
Cidade Residencia	Campina Grande	Campina Grande	Campina Grande	Campina Grande	Campina Grande
Data Nascimento	16/6/1981	28/8/1985	30/1/1996	30/8/1978	17/4/1982
CEP	58.000-000	58.415-105	58.414-285	58.433-181	58.475-000
Bairro	CATOLE	JARDIM PAULISTANOSS	LIBERDADE	MALVINAS	CENTRO

Fonte: Autoria própria, 2021

Na Figura 18, apresenta-se a *Pipeline* de transformação para limpeza dos dados brutos obtidos. O primeiro estágio de transformação, o filtro por cidade, elimina as entradas que especificam pacientes de cidades fora de Campina Grande. O segundo estágio retira entradas cujo paciente apresentou resultado negativo para a doença. O último estágio retira da tabela entradas que possuem valores nulos para as colunas Data PCR ou Data Sorológico, já que sem esses valores não é possível determinar o dia em que o paciente foi identificado com a doença. Essa retirada é importante já que estamos lidando com

séries temporais, dessa forma, é imprescindível que mantenhamos um acompanhamento das datas de identificação de pacientes.

Figura 18 – *Pipeline* de transformação dos dados brutos

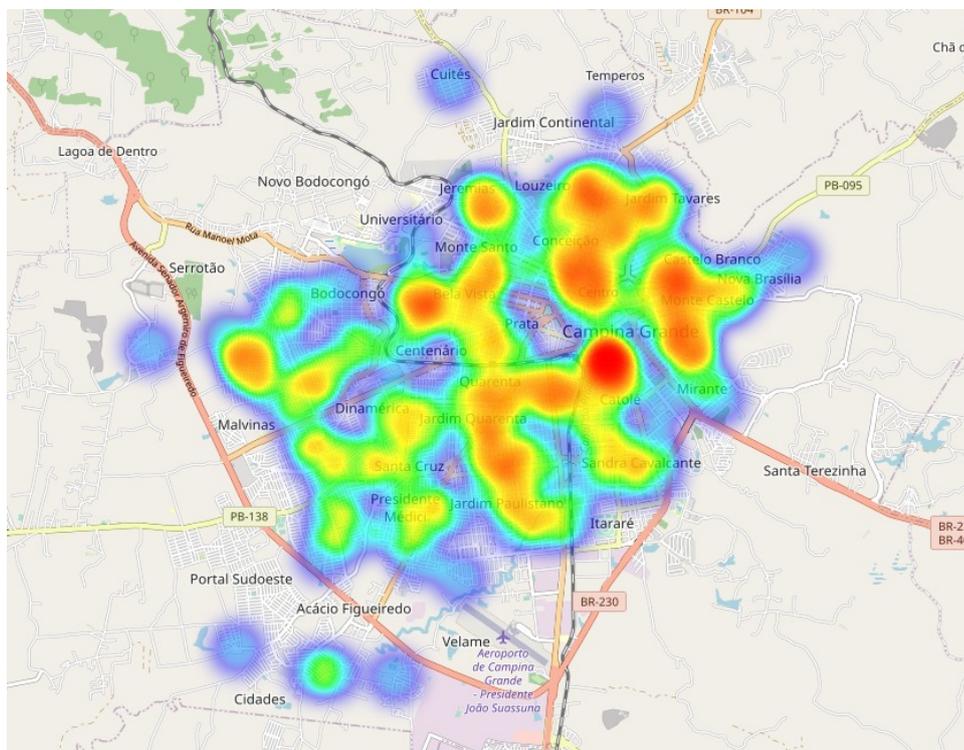


Fonte: Autoria própria, 2021

Para a obtenção da quantidade total de casos diários, realizou-se uma operação de agrupamento da tabela por data e em seguida uma agregação por soma de casos diários. Para a obtenção dos acumulados de casos ao longo do tempo realizou-se uma operação de soma cumulativa do total de casos diários.

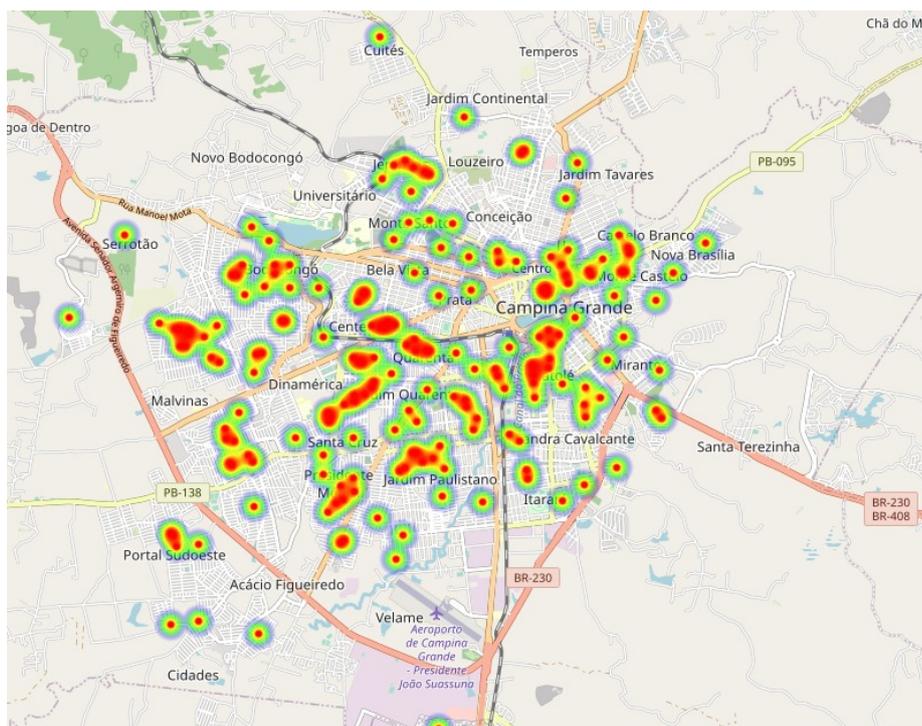
Em seguida, para fins de visualização, utilizamos a coluna de CEP para identificar as coordenadas geográficas do bairro de cada paciente. Isso permite um acompanhamento geográfico de casos de Covid na cidade de Campina Grande e pode ser utilizado em trabalhos futuros envolvendo agrupamentos. Pôde-se construir um mapa de calor que evolui ao longo do tempo contendo o acumulado de casos de Covid e a quantidade de casos ativos na cidade. As Figuras 19 e 20 trazem uma amostra desses mapas de calor no início de Junho de 2020. A Figura 19 mostra o acumulado de casos e a figura 20 mostra a quantidade de casos ativos no dia. Podemos perceber que as regiões centrais da cidade são as mais afetadas pela infecção.

Figura 19 – Mapa de calor da quantidade acumulada de casos por bairro em Campina Grande no início de Junho de 2020



Fonte: Autoria própria, 2021

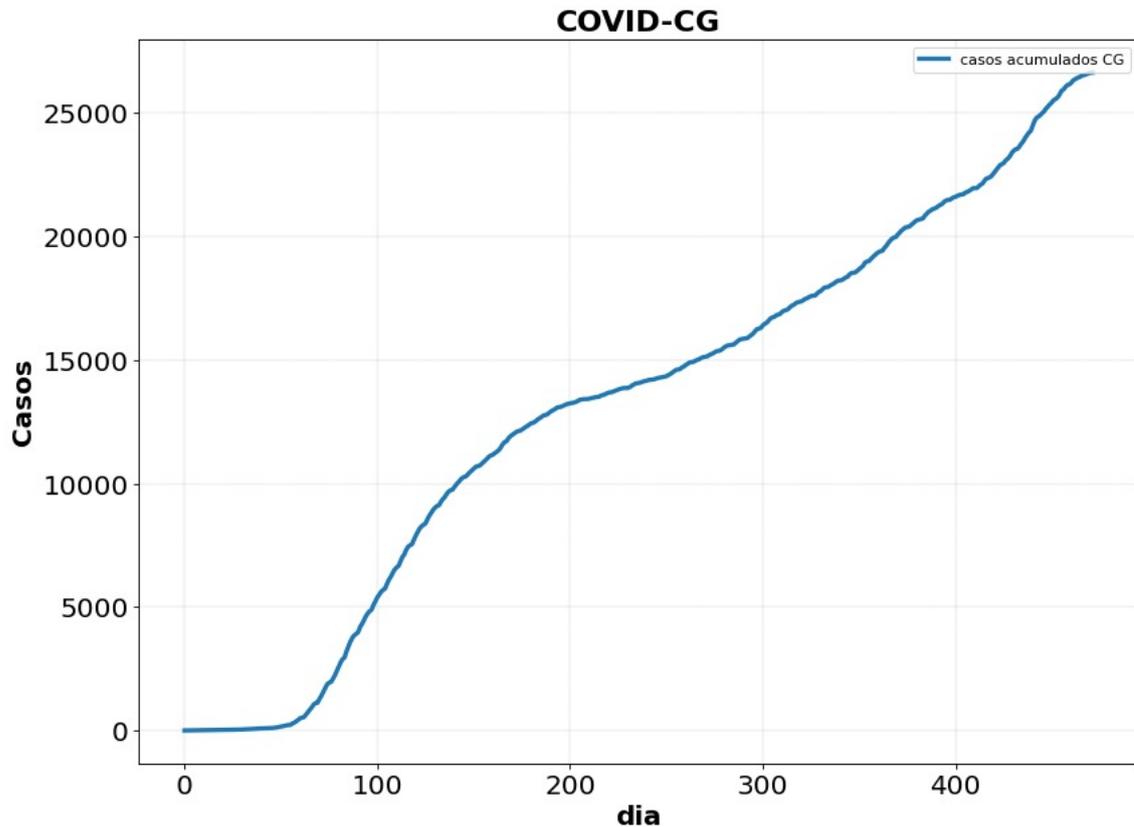
Figura 20 – Mapa de calor da quantidade ativa de casos por bairro em Campina Grande no início de Junho de 2020



Fonte: Autoria própria, 2021

O período de análise e modelagem dos dados compreende o período entre o dia 02 de fevereiro de 2020 e o dia 01 de julho de 2021, totalizando 472 dias. O gráfico do acumulado de casos ao longo do tempo pode ser observado na Figura 21

Figura 21 – Evolução do acumulado de casos de Covid em Campina Grande - PB



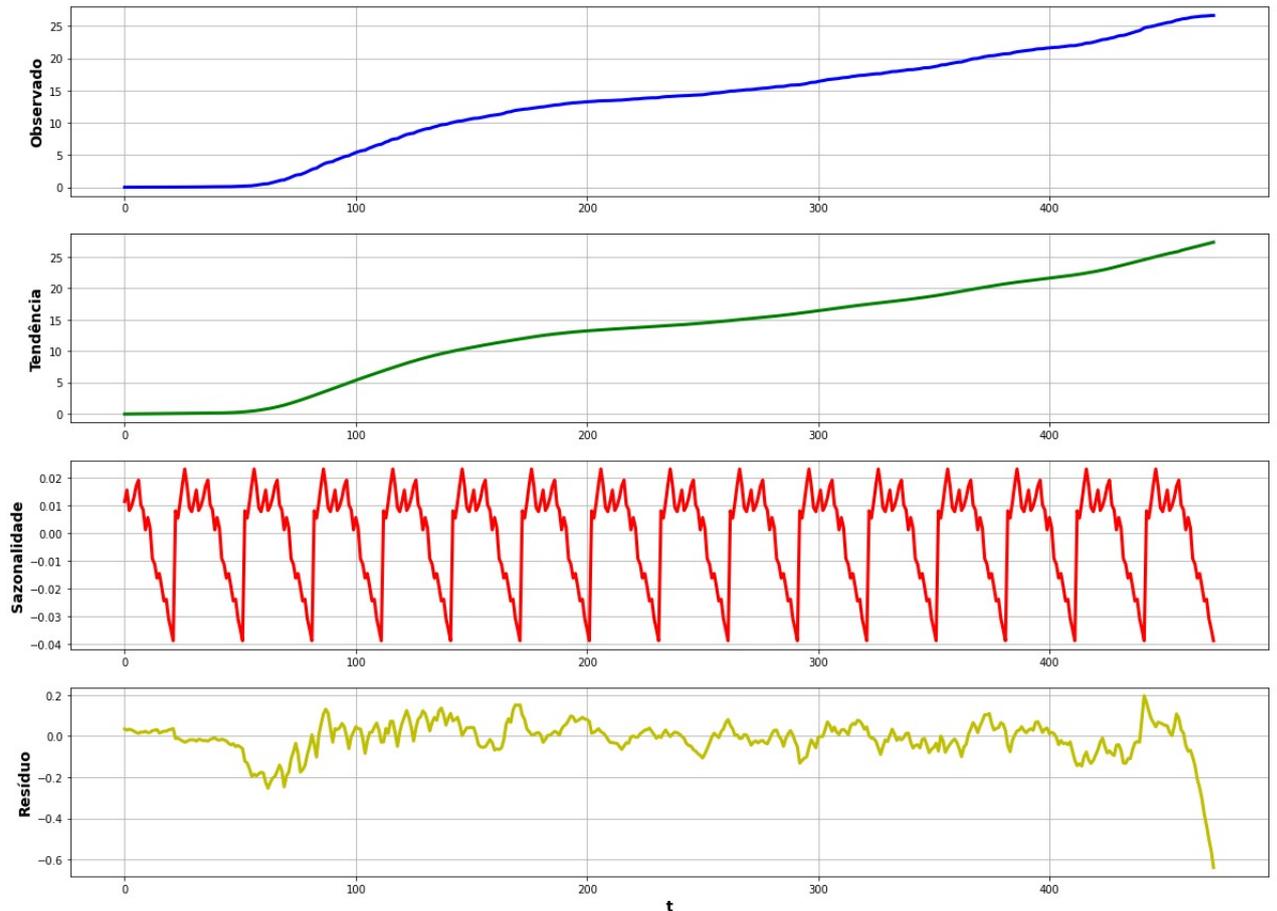
Fonte: Autoria própria, 2021

Qualitativamente, podemos observar que trata-se de uma série temporal não estacionária com uma forte componente de tendência. De fato, pela própria característica do cumulativo de casos, temos uma função monotonicamente crescente. Na Figura 22, podemos avaliar a decomposição dessa série, de maneira análoga às sessões anteriores. A componente de sazonalidade parece ser fraca em relação à componente de tendência já que possui uma ordem de grandeza muito inferior.

3.4.1 Modelagem do problema

Obtido o conjunto de dados que representa a o acumulado de casos ao longo do tempo, precisamos definir uma estrutura para a modelagem do problema, isso é, definir uma variável alvo e um conjunto de variáveis regressoras. Para este problema específico, definimos como a variável alvo o acumulado de casos no dia $d + 7$, dessa forma estaremos prevendo a evolução da curva 7 dias para o futuro.

Figura 22 – Decomposição aditiva da série temporal do acumulado de casos

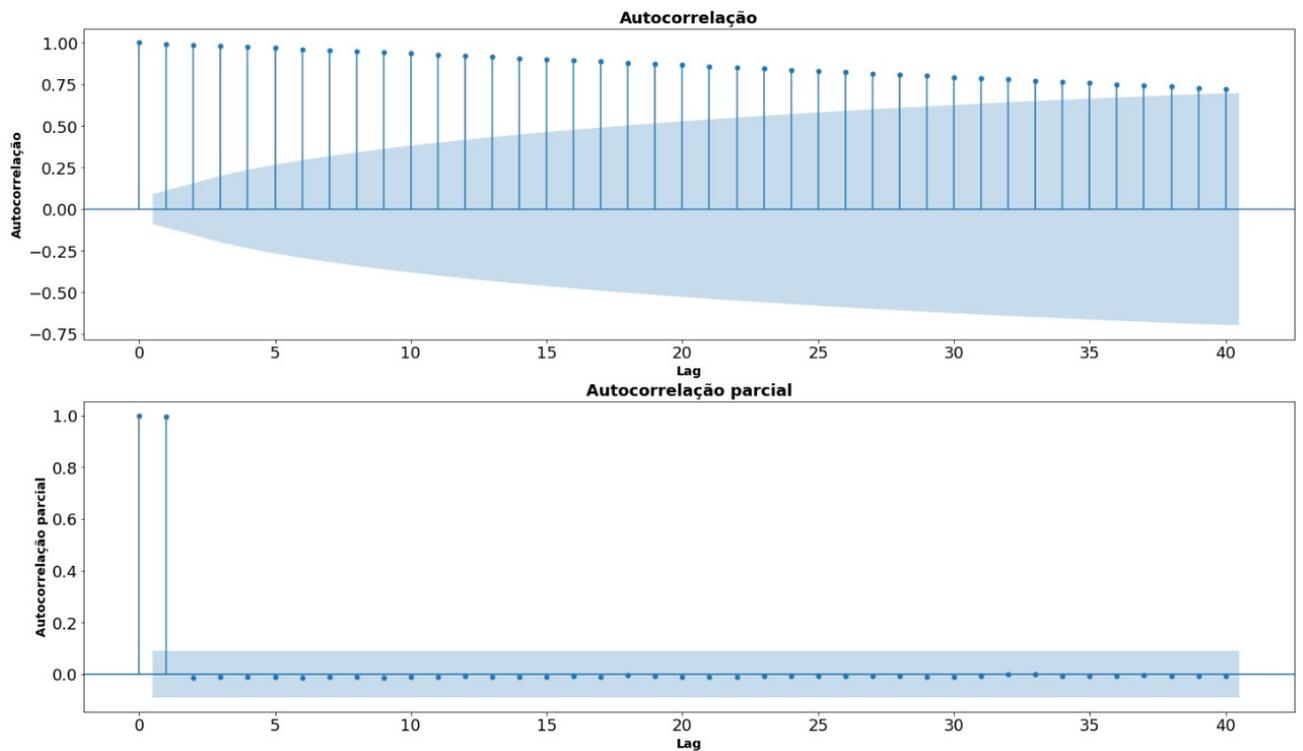


Fonte: Autoria própria, 2021

O conjunto de variáveis regressoras foi definido utilizando apenas valores passados da série. É importante notar que para modelos como o ARIMA, é importante analisar a série de maneira estacionária para definir a quantidade de *lags* das variáveis autorregressoras. Isso implica em realizar um estudo da estacionariedade da série, além de analisar os componentes de autocorrelação e autocorrelação parcial. Como estamos utilizando um modelo baseado em aprendizagem de máquina, podemos automatizar o processo de escolha de variáveis e decidir quais são as parcelas mais significativas para um bom ajuste dos dados.

Dito isso, observou-se o gráfico de autocorrelação e de autocorrelação parcial da série não estacionária (Figura 23) e decidiu-se utilizar como variáveis regressoras, a média móvel do acumulado de casos no conjunto de dias $\{d-7, d-6, \dots, d\}$ e a própria quantidade de casos no dia d .

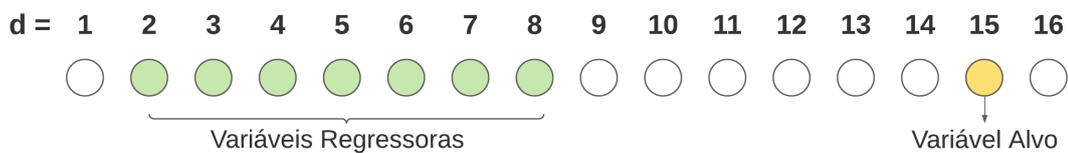
Figura 23 – Autocorrelação e autocorrelação parcial série temporal do acumulado de casos



Fonte: Autoria própria, 2021

A estrutura de variáveis utilizadas pode ser observada na Figura 24

Figura 24 – Modelagem de variáveis utilizadas para o acumulado de casos



Fonte: Autoria própria, 2021

3.4.2 Estruturas de *kernels* proposta

Dado a análise inicial do problema, podemos supor que uma estrutura de *kernel* para regressão polinomial ou para a abordagem semi-paramétrica, conforme mostrado na Tabela 1, seja adequada para capturar a tendência dessa série temporal. Dessa maneira, foi implementada uma busca em grade (do inglês, *grid search*) para encontrar as melhores combinações de parâmetros dos kernels.

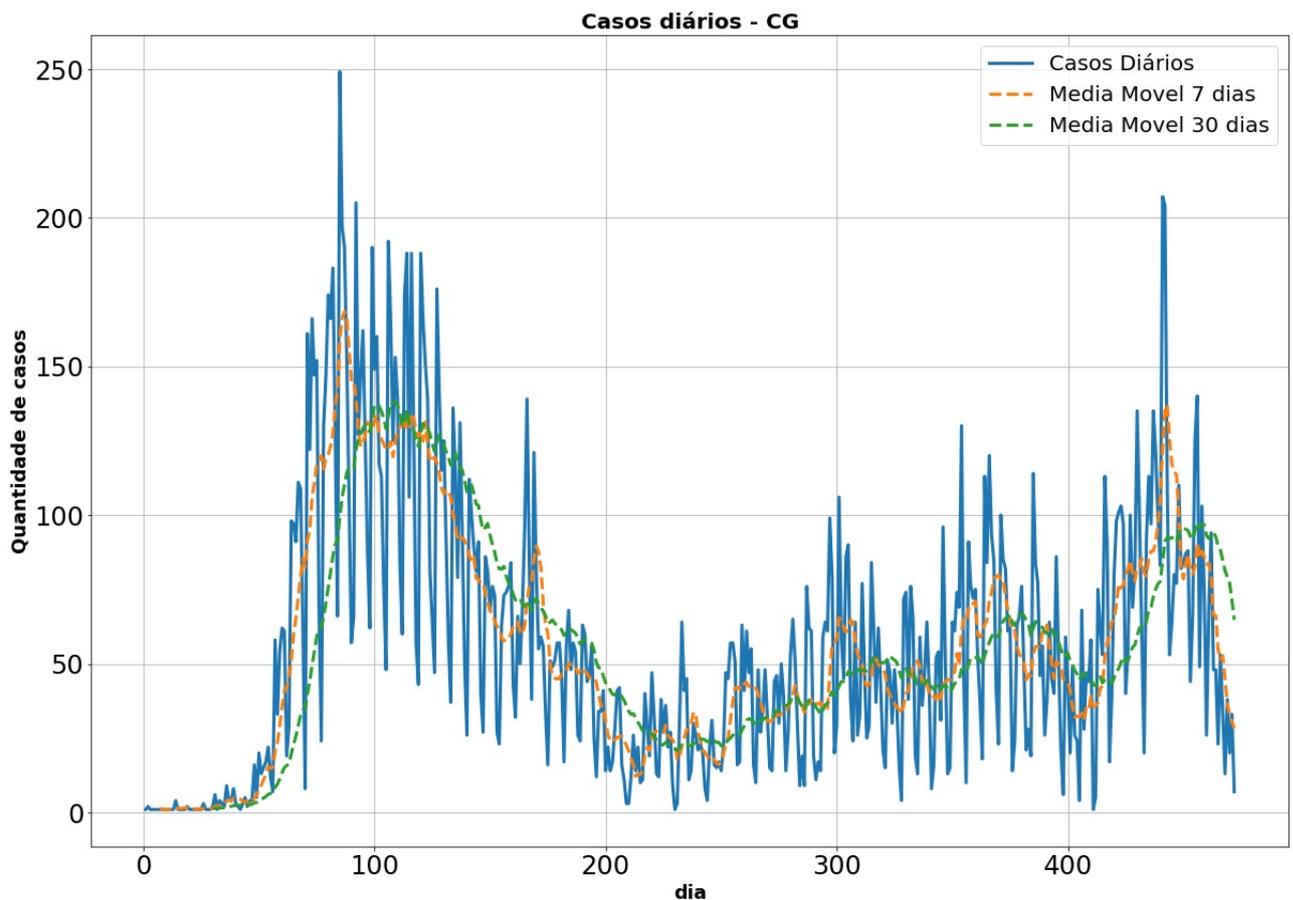
Para esse caso, propomos uma variação do valor σ do *kernel* linear dentro do conjunto $\{1, 2, 3, \dots, 10\}$ e uma variação do *lengthscale* do *kernel* SE entre o espaço logarítmico de $[-1, 1]$ coletando 5 amostras igualmente espaçadas.

Os resultado das métricas r^2 , MSE e MAPE para os melhores *kernels* encontrados são mostrados na seção 4 deste documento. Além disso, apresenta-se a previsão e o intervalo de confiança para o melhor modelo treinado. Apresenta-se, também, a distribuição dos erros de predição no conjunto de teste.

3.5 Previsão da média móvel de casos de Covid

Os dados obtidos para a quantidade de casos novos diários podem ser observados na Figura 25. Podemos ver, também, a média móvel de casos novos com uma janela de 7 dias e com uma janela de 30 dias.

Figura 25 – Evolução dos casos diários



Fonte: Autoria própria, 2021

Diferentemente do acumulado de casos, a previsão da quantidade diária de casos de Covid-19 é um problema complexo e desafiador. Seja qual for a modelagem utilizada

— modelo ARIMA, redes neurais, GPR — podemos supor que haverá dificuldade de se manter bons resultados em conjuntos de testes devido à natureza imprevisível da evolução da quantidade diária de casos. Notícias na televisão, medidas de quarentena, quantidade de vacinação ou próprios erros de aquisição dos dados afetam fortemente a curva que observamos na Figura 25. Isso se reflete em uma variância muito forte dos dados: a média de casos diários durante todo o período observado foi de 56,44, enquanto a variância foi de 46,75. Diante disso, de modo a obter uma variável alvo mais suave, utilizamos a média móvel de casos com uma janela de 7 dias, ilustrada por meio da linha amarela tracejada na Figura 25.

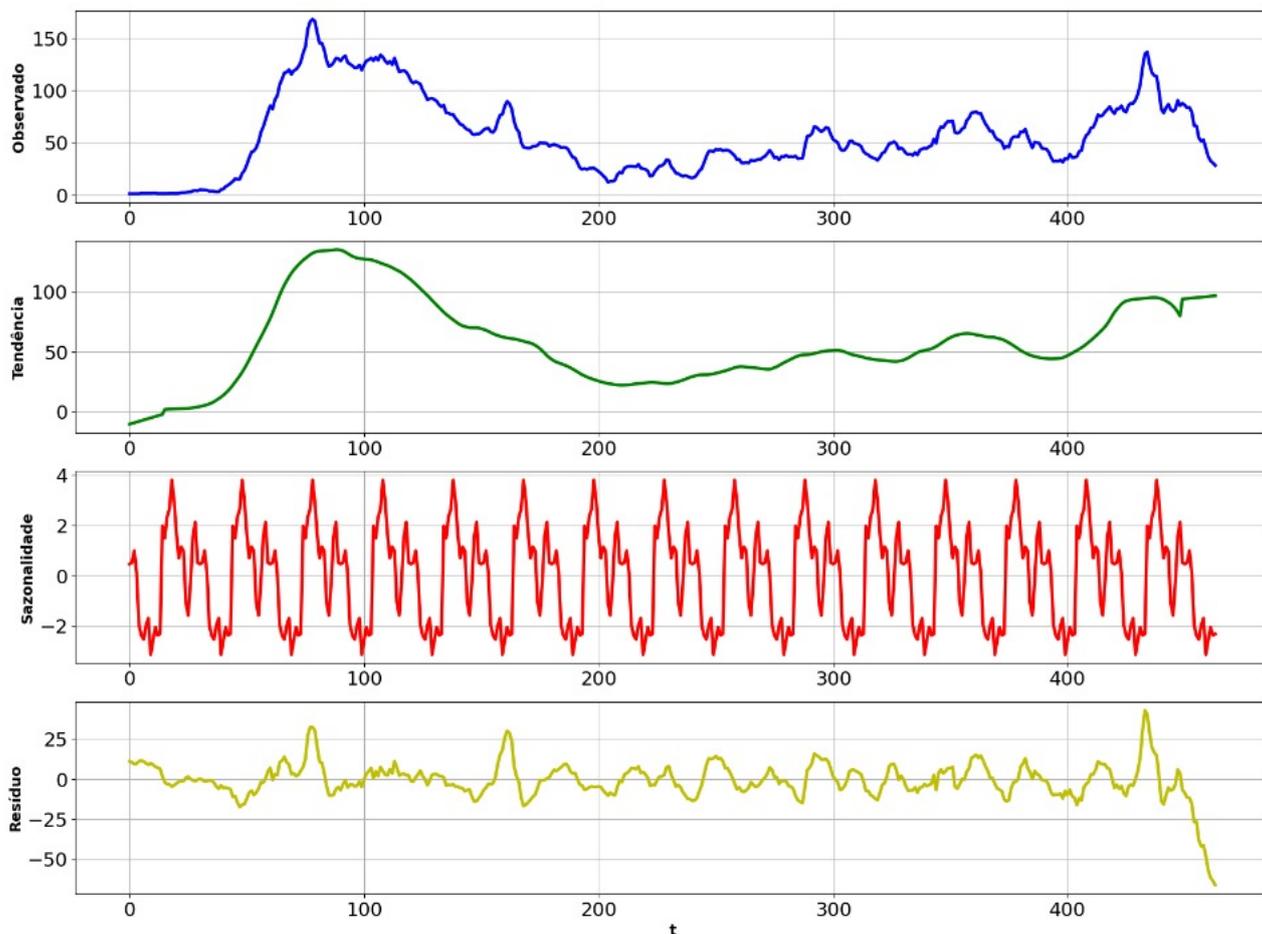
De maneira similar aos problemas anteriores, o desenvolvimento dos modelos iniciou-se com a análise da decomposição da série temporal da variável alvo, ilustrado na Figura 26. Conforme podemos visualizar, há uma mudança na tendência da série ao longo do tempo: aproximadamente nos 90 primeiros dias de contágio da doença houve um rápido crescimento na média móvel de novos casos; em seguida há uma queda até os dia 200 da pandemia e, então, há um crescimento mais lento até o final das observações. Já a componente de sazonalidade apresenta uma ordem de grandeza inferior. Pode-se perceber que embora haja uma repetição padronizada mensal de novas infecções, a componente é menos significativa que a componente de tendência. A decomposição também indicou um forte componente de resíduo, com valores que superam a parcela de sazonalidade. Esse é um outro indicativo da natureza irregular da série temporal com a qual lidamos.

3.5.1 Modelagem do problema

De maneira análoga à anterior, precisamos definir o conjunto de variáveis regressoras (ou *features*) e a variável alvo para a utilização do GPR. A variável alvo foi definida como a média móvel de casos (com janela de 7 dias) 7 dias para o futuro. Isso é, no dia d desejamos prever a média móvel de casos da semana seguinte. A previsão de janelas de tempo menores é um problema ainda mais difícil devido à maior variância da série. Quanto maior a janela de tempo de previsão, mais suave deve ser a variável e, conseqüentemente, mais simples se torna a tarefa de regressão.

Para a definição do conjunto de variáveis regressoras, utilizou-se o mesmo procedimento de análise dos gráficos de correlação e autocorrelação parcial, apresentados na Figura 27. Nota-se que existe uma forte autocorrelação entre os primeiros *lags* da série e que essa influência decai gradualmente ao longo dos *lags*. Essa é uma característica esperada de uma série não estacionária formada por uma média móvel. É importante destacar a influência da autocorrelação parcial do *lag* de índice 8. Como a variável alvo é a média móvel de casos no dia $d + 7$, no dia d não temos acesso às componentes $d + 1, d + 2, \dots, d + 6$. Dessa forma, podemos utilizar apenas as componentes do conjunto $d, d - 1, d - 2, \dots$ para a construção de *lag features*. Para esse problema específico, utilizou-se como conjunto de

Figura 26 – Decomposição aditiva da série temporal dos casos diários



Fonte: Autoria própria, 2021

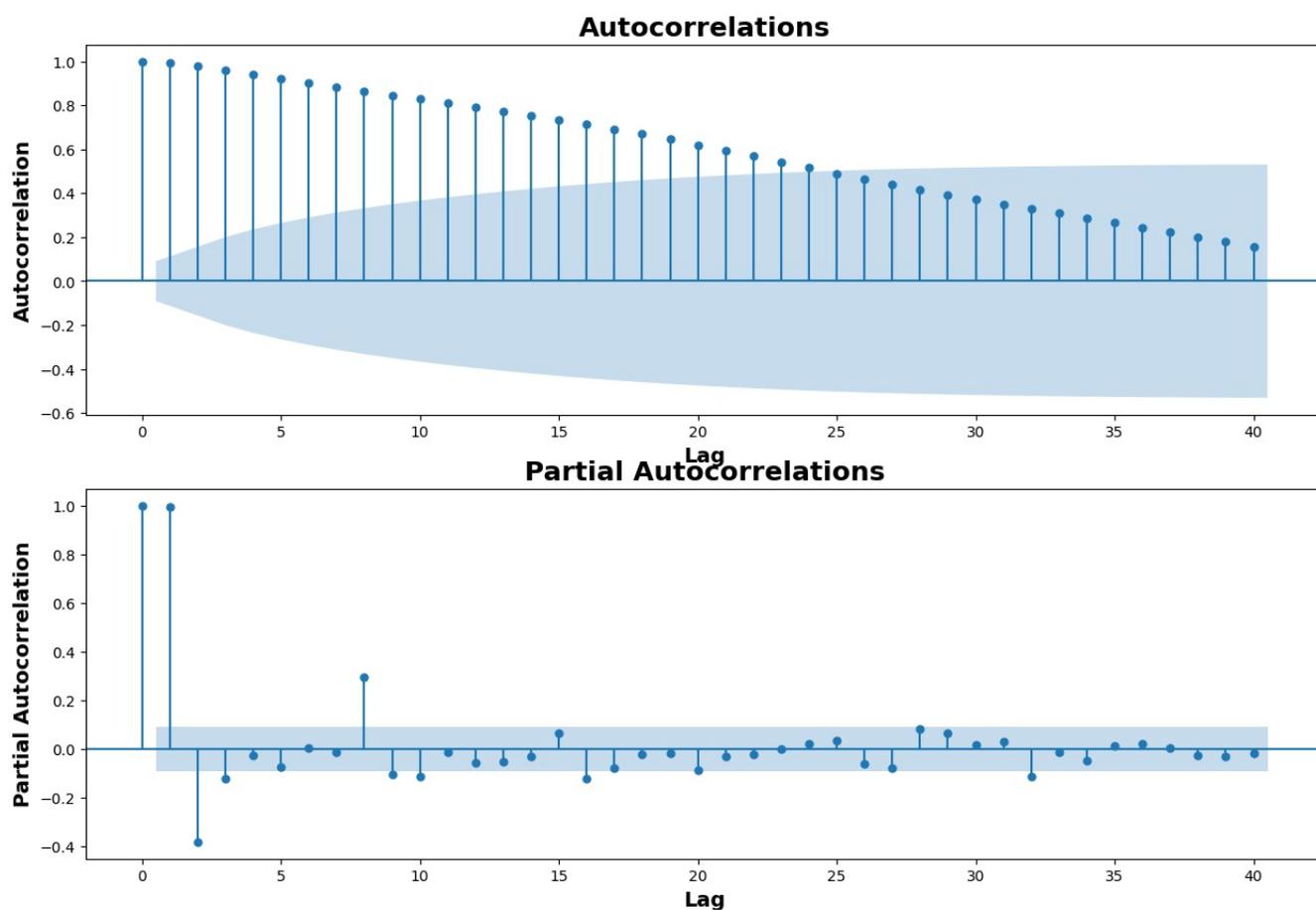
variáveis regressoras o tempo, a quantidade de casos diários no dia d e os $lags_{1,2,3,4,5,6,7}$ da variável alvo.

3.5.2 Estruturas de *kernels* proposta

Diante do cenário descrito, foram propostas duas estruturas de *kernels*: a estrutura de regressão semi-paramétrica e a estrutura *Trend, cyclical, irregular*, conforme apresentado na Tabela 1. Também foi implementada uma busca em grade para encontrar as melhores combinações de parâmetros dos *kernels* dentro do espaço de busca definido.

Para a estrutura semi-paramétrica, uma pequena modificação foi proposta. Em vez da utilização de apenas um *kernel* SE, utilizou-se o *kernel* conhecido como racional quadrático (do inglês *rational quadratic* ou RQ), equivalente à soma de vários *kernels* SE com diferentes *lengthscales* (DUVENAUD, 2014). Isso foi feito como uma tentativa de capturar as diferentes formas de variações observadas na série ao longo do tempo. A

Figura 27 – Autocorrelação e autocorrelação parcial série temporal da média móvel dos casos diários



Fonte: Autoria própria, 2021

seguinte estrutura de código foi implementada para essa estrutura:

```

1 # Regressao semi parametrica
2
3 semi_parametric_kernels = []
4
5 ## Linear -- N o estacionario
6 linear_kernels = [DotProduct(**exponent for exponent in range(1,3)]
7
8
9 # SE - estacionario
10 sqr_exp_kernels = [RationalQuadratic(l) for l in [1,10,100]]
11
12
13 noise_kernels = [WhiteKernel(noise_level=nl) for nl in np.arange
14                 (0.1,5,0.6)]
15
16 for linear_kernel in linear_kernels:

```

```

17     for squared_exponential_kernel in sqr_exp_kernels:
18         for noise_kernel in noise_kernels:
19             semi_parametric_kernels.append(linear_kernel +
squared_exponential_kernel + noise_kernel)

```

Dentro dessa estrutura, precisamos destacar o espaço de busca de parâmetros considerados. Para o *kernel* linear, propomos uma variação do expoente da ordem 1 até a ordem 3. Para o *kernel* RQ, variamos o valor do *lengthscale* no espaço do conjunto 1, 10, 100 e para o *kernel* de ruído, variamos o nível do ruído no espaço de 0.1 à 5 com um passo de 0.6. Ao todo, definiu-se um espaço de busca com 54 *kernels* diferentes.

Para a estrutura *Trend, cyclical, irregular*, composta por uma somatória de *kernels* SE (equivalente ao *kernel* RQ), por uma somatória de *kernels* periódicos e por um *kernel* de ruído, outro espaço de busca foi proposto (mostrado no trecho de código a seguir). A equação 3.5 apresenta a estrutura desse *kernel*.

$$Structure = \sum SE + \sum periodico + WN \quad (3.5)$$

```

1 # Trend, cyclical, irregular
2
3 trend_cyclical_kernels = []
4
5 ## Linear -- N o estacion rio
6 exp_sine_kernels = [ExpSineSquared(periodicity=p) for p in
    [1,3,7,10,15,30]]
7
8 #squared exponential
9
10 rational_quadratic_kernels = [RationalQuadratic(l) for l in [1,10,100]]
11
12 noise_kernels = [WhiteKernel(noise_level=nl) for nl in np.arange
    (0.1,5,0.6)]
13
14
15 for esk in exp_sine_kernels:
16     for rqk in rational_quadratic_kernels:
17         for nk in noise_kernels:
18             trend_cyclical_kernels.append(esk + rqk + nk)

```

O espaço de busca possui algumas semelhanças com o anterior: os mesmos parâmetros foram definidos para o conjunto de *kernels* RQ e para os *kernels* de ruído. Já para o *kernel* periódico, variou-se o valor da periodicidade no conjunto {1, 3, 7, 10, 15, 30}. Ao todo, definiu-se um espaço de busca com 162 *kernels* diferentes.

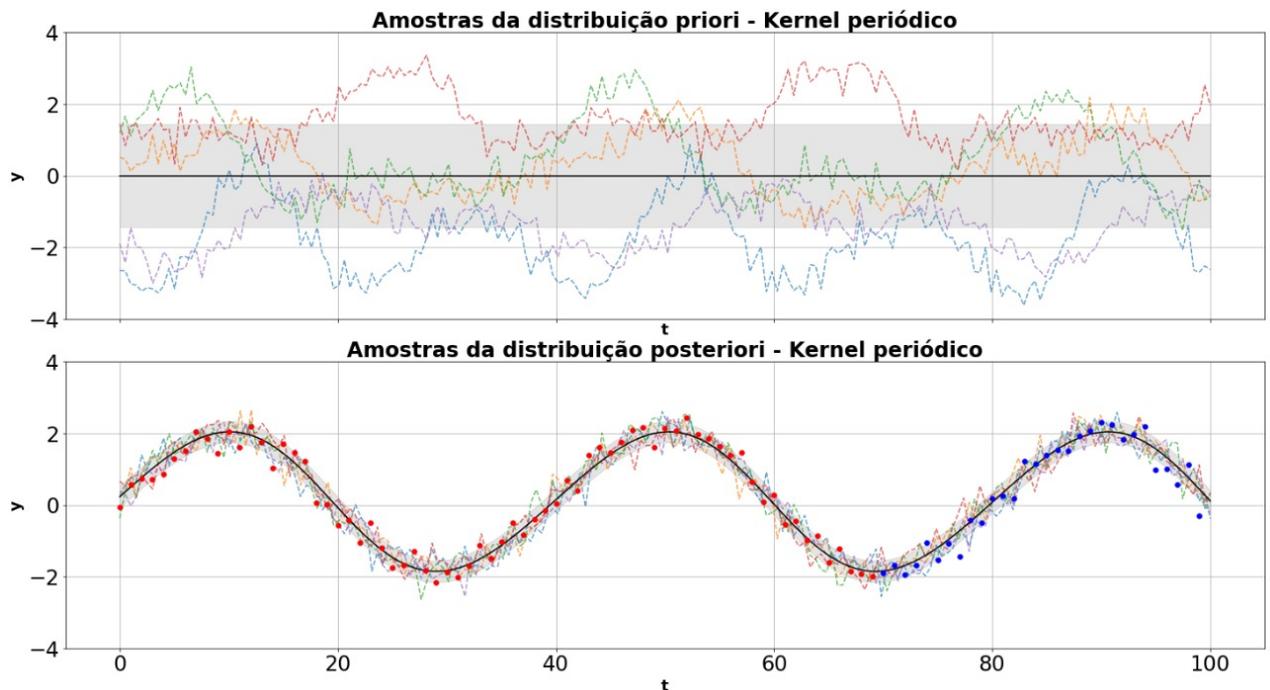
4 Resultados Obtidos

Nesta seção, apresentamos os resultados obtidos para as análises propostas na Metodologia.

4.1 Série temporal apenas com componente de sazonalidade

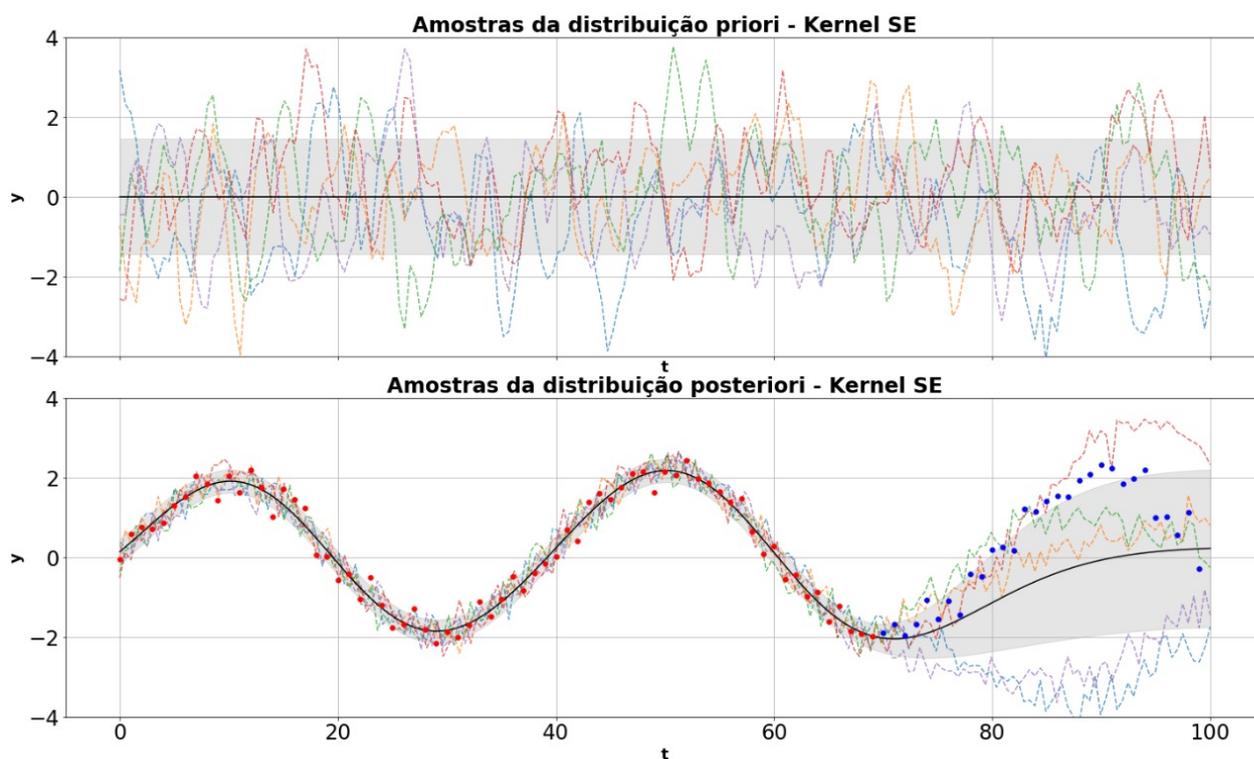
Para o primeiro problema base proposto, podemos observar nas Figuras a seguir amostras das distribuições prioris e posterioris para os *kernels* discutidos na seção 3.1.2. Para esse tipo de Figura, temos nas cores azul, amarela, verde, vermelha e roxa exemplos de funções amostras. É importante destacar que, embora apenas 5 amostras sejam mostradas no gráfico, inúmeras outras poderiam ser apresentadas. Na linha preta cheia temos a média da previsão, tanto para os gráficos a priori quanto para os a posteriori. A região cinza indica o intervalo de confiança de ± 1 desvio padrão. Os círculos vermelhos e azuis indicam os dados de treino e de teste, respectivamente, utilizados no processo de treinamento dos modelos.

Figura 28 – Amostras a priori e a posteriori do *kernel* periódico para a série sazonal



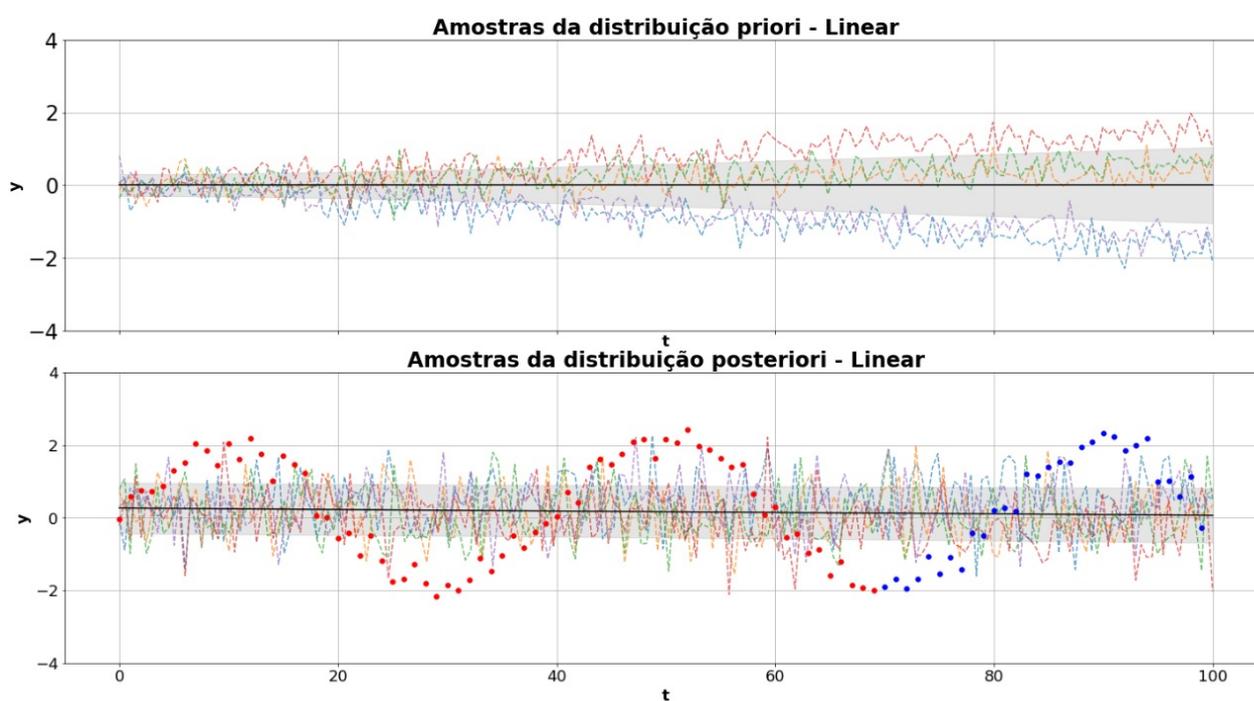
Fonte: Autoria própria, 2021

Figura 29 – Amostras a priori e a posteriori do *kernel* SE para a série sazonal



Fonte: Autoria própria, 2021

Figura 30 – Amostras a priori e a posteriori do *kernel* linear para a série sazonal

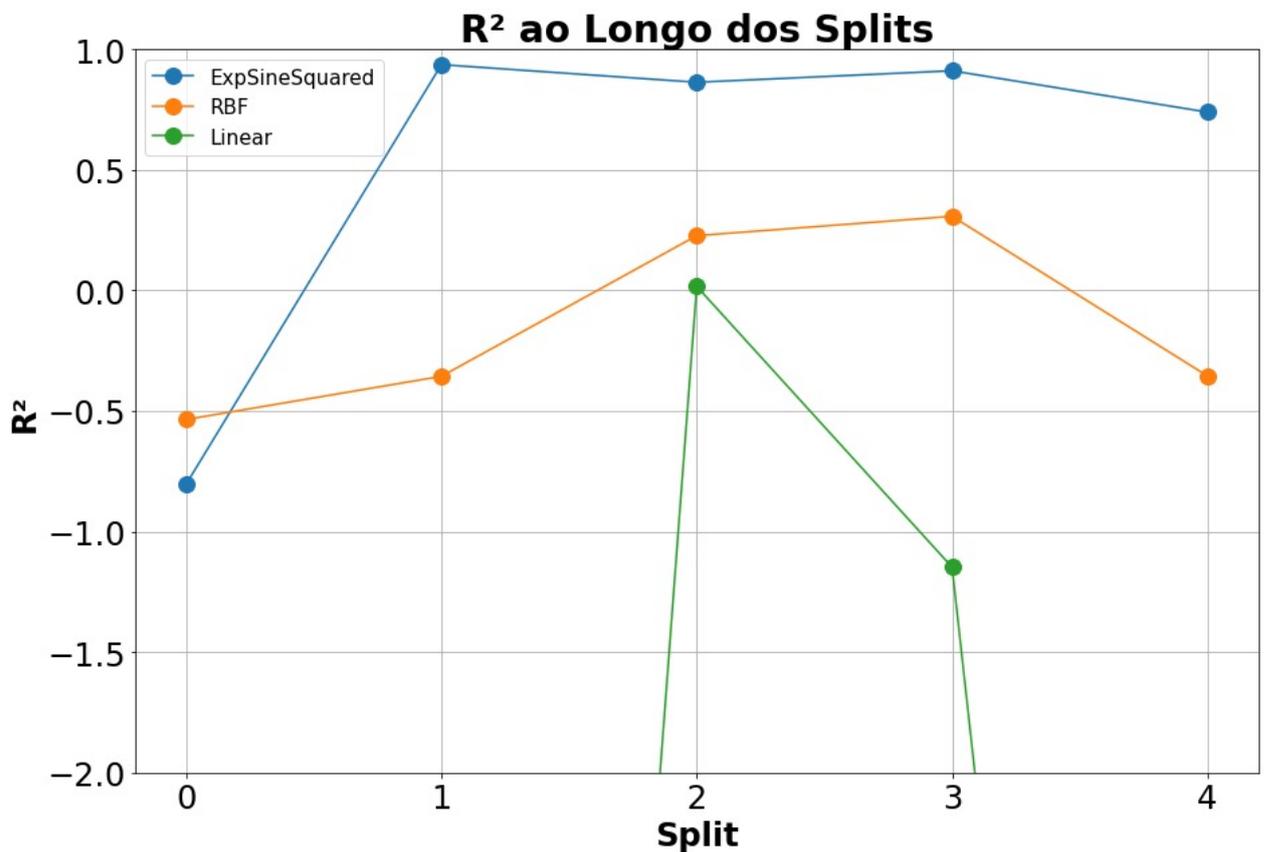


Fonte: Autoria própria, 2021

Qualitativamente, podemos perceber, assim como supomos, que as previsões para o *kernel* periódico superam bastante as outras duas estruturas de *kernel*. Isso se dá devido à natureza sazonal da série temporal considerada. Embora o *kernel* SE tenha sido capaz de se ajustar ao conjunto de treino, não obteve um bom resultado no conjunto de teste, indicando que houve um *overfit*. O *kernel* linear foi incapaz de realizar um ajuste adequado e acabou alocando todas as previsões na média do fenômeno.

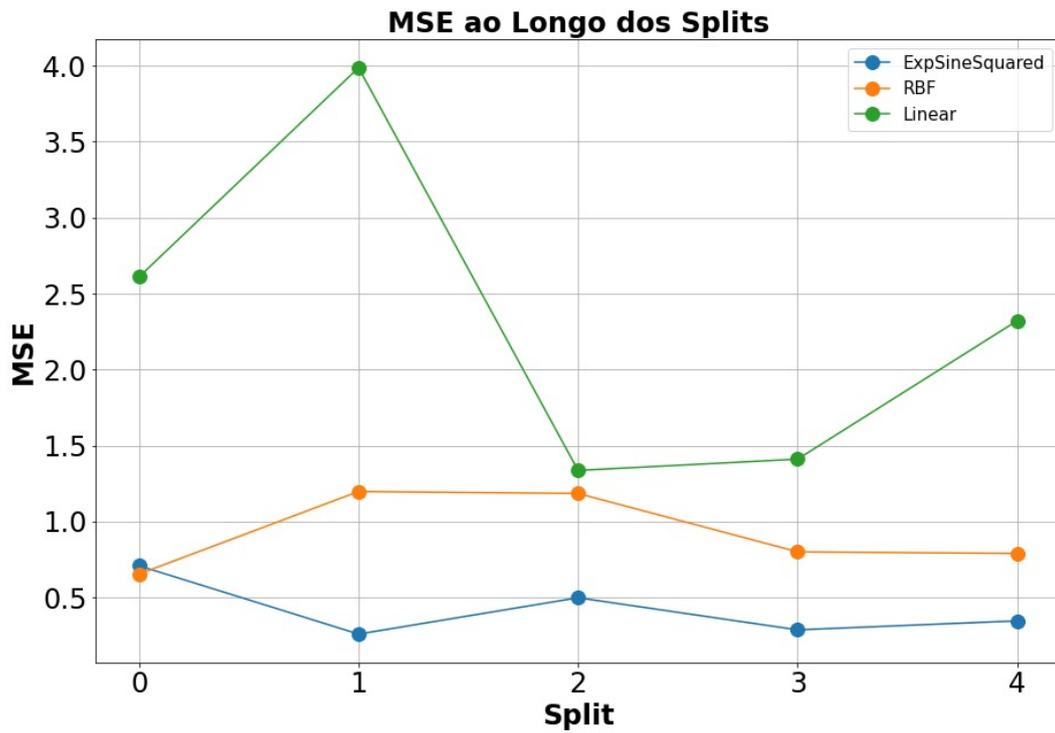
Uma comparação quantitativa das métricas propostas para esses 3 *kernels* ao longo dos *splits* da validação cruzada pode ser observada nas Figuras 31, 32 e 33. Novamente podemos verificar uma superioridade nos resultados do *kernel* periódico. No gráfico da Figura 31 vemos que o valor do r^2 do modelo periódico se mantém próximo a 1 desde os *splits* iniciais.

Figura 31 – Comparação do R^2 ao longo dos splits - apenas sazonalidade



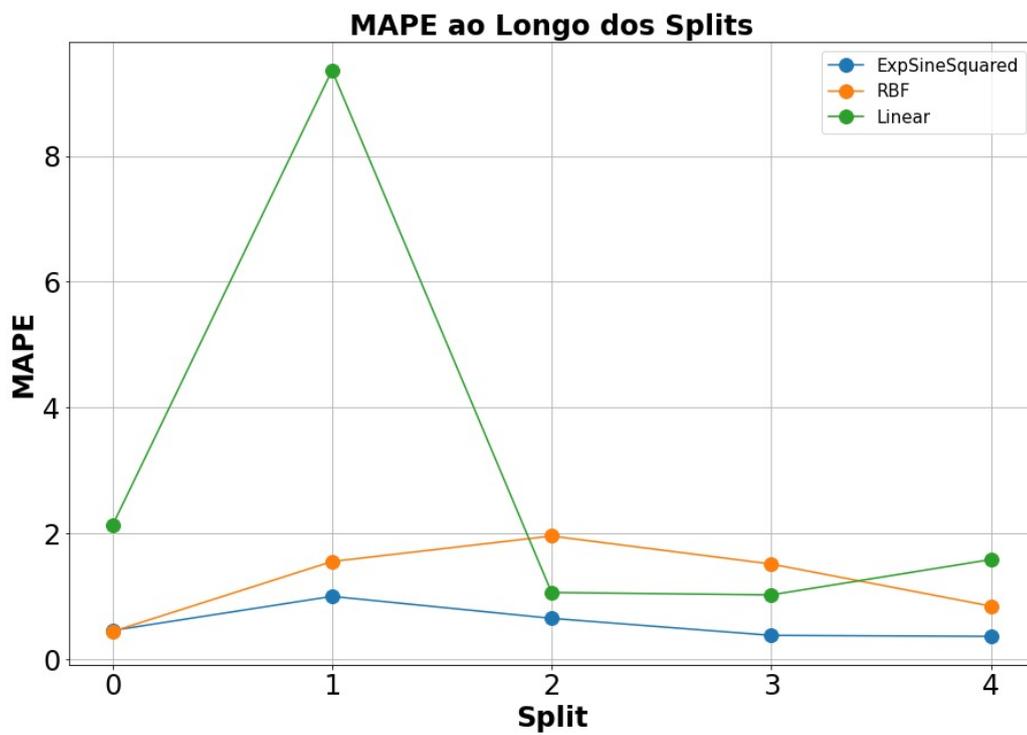
Fonte: Autoria própria, 2021

Figura 32 – Comparação do MSE ao longo dos splits - apenas sazonalidade



Fonte: Autoria própria, 2021

Figura 33 – Comparação do MAPE ao longo dos splits - apenas sazonalidade

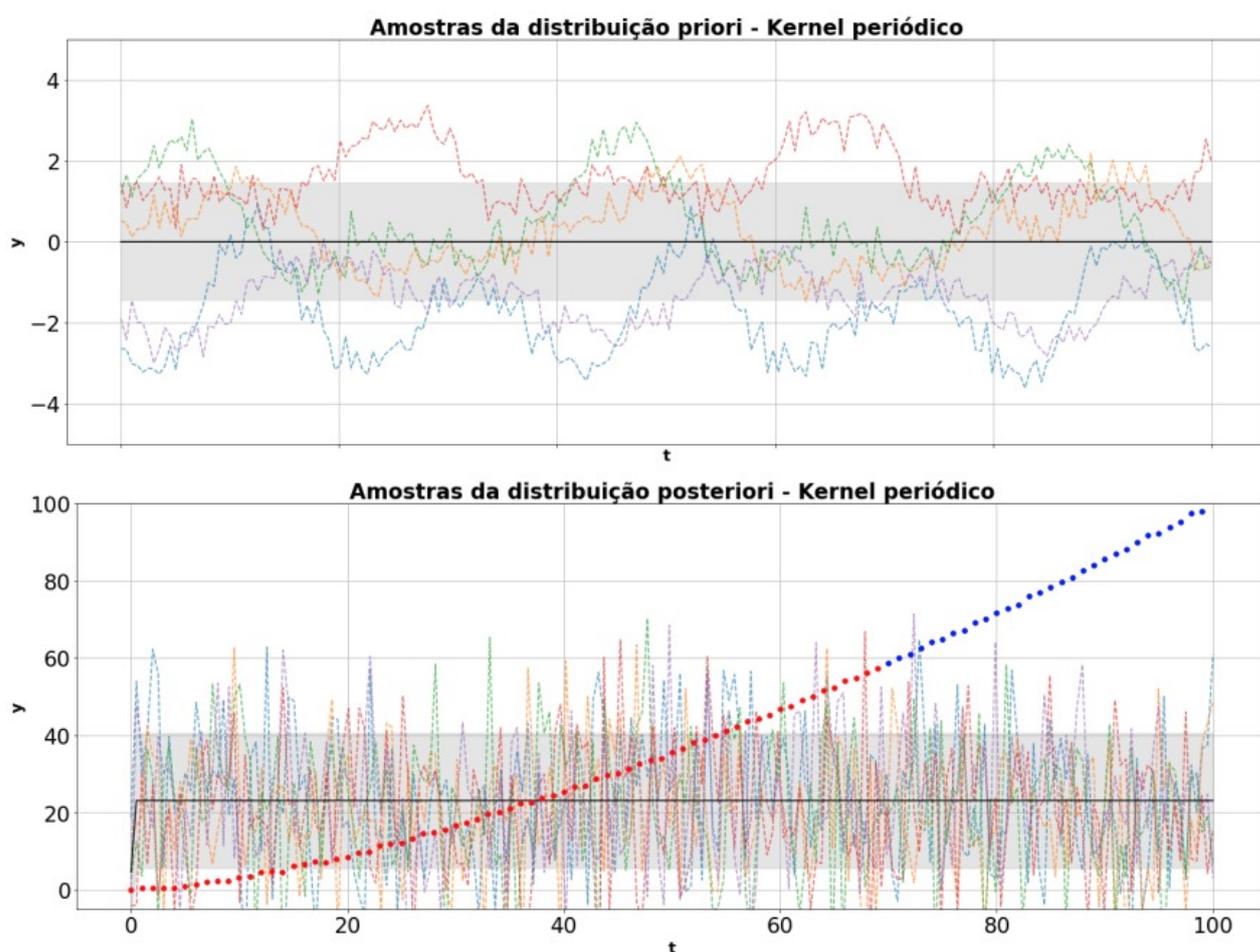


Fonte: Autoria própria, 2021

4.2 Série temporal apenas com componente de tendência

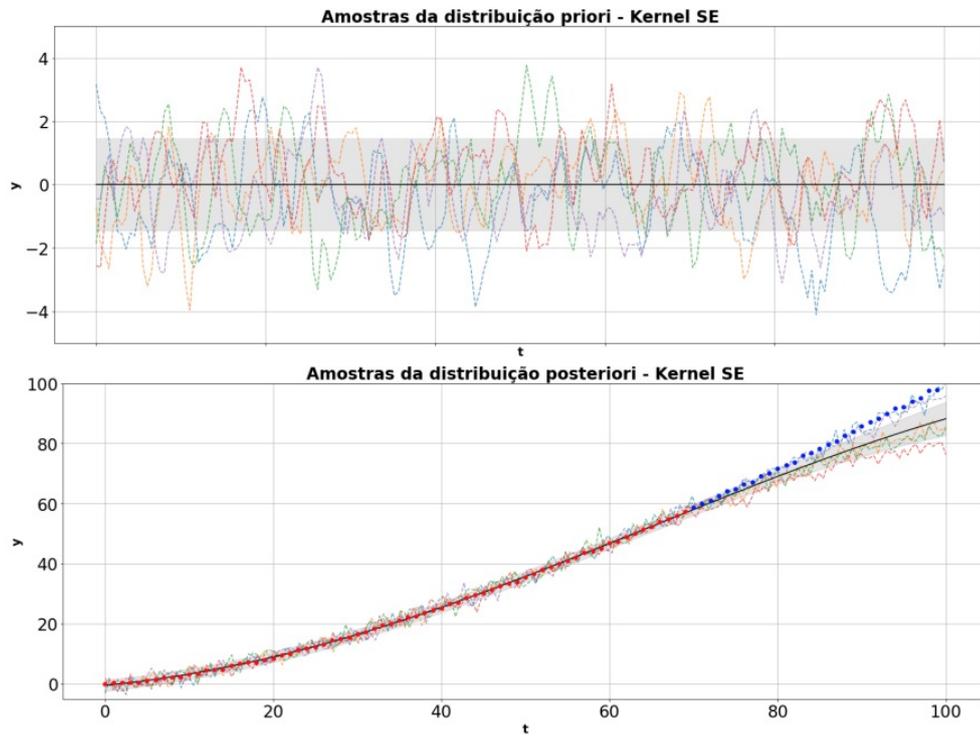
Para o segundo problema base proposto, podemos observar nas Figuras a seguir amostras das distribuições prioris e posterioris para os *kernels* discutidos na seção 3.2.2. Novamente, para esse tipo de Figura, temos nas cores azul, amarela, verde, vermelha e roxa exemplos de funções amostras. Na linha preta cheia temos a média da previsão, tanto para os gráficos a priori quanto para os a posteriori. A região cinza indica o intervalo de confiança de ± 1 desvio padrão. Os círculos vermelhos e azuis indicam os dados de treino e de teste, respectivamente, utilizados no processo de treinamento dos modelos.

Figura 34 – Amostras a priori e a posteriori do *kernel* periódico para a série com tendência



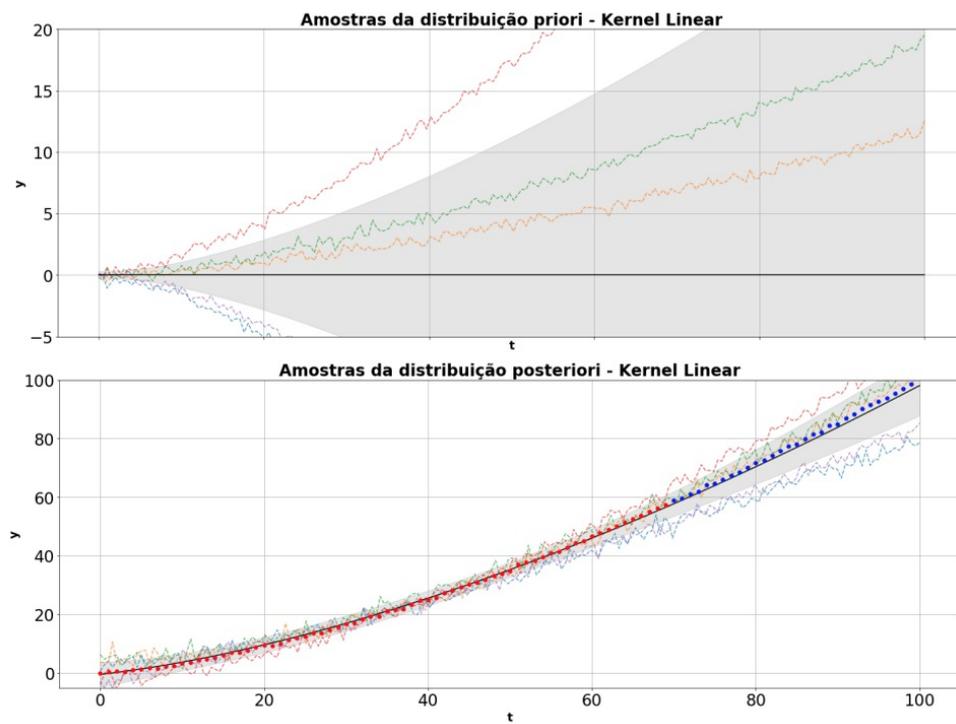
Fonte: Autoria própria, 2021

Figura 35 – Amostras a priori e a posteriori do *kernel* SE para a série com tendência



Fonte: Autoria própria, 2021

Figura 36 – Amostras a priori e a posteriori do *kernel* linear para a série com tendência

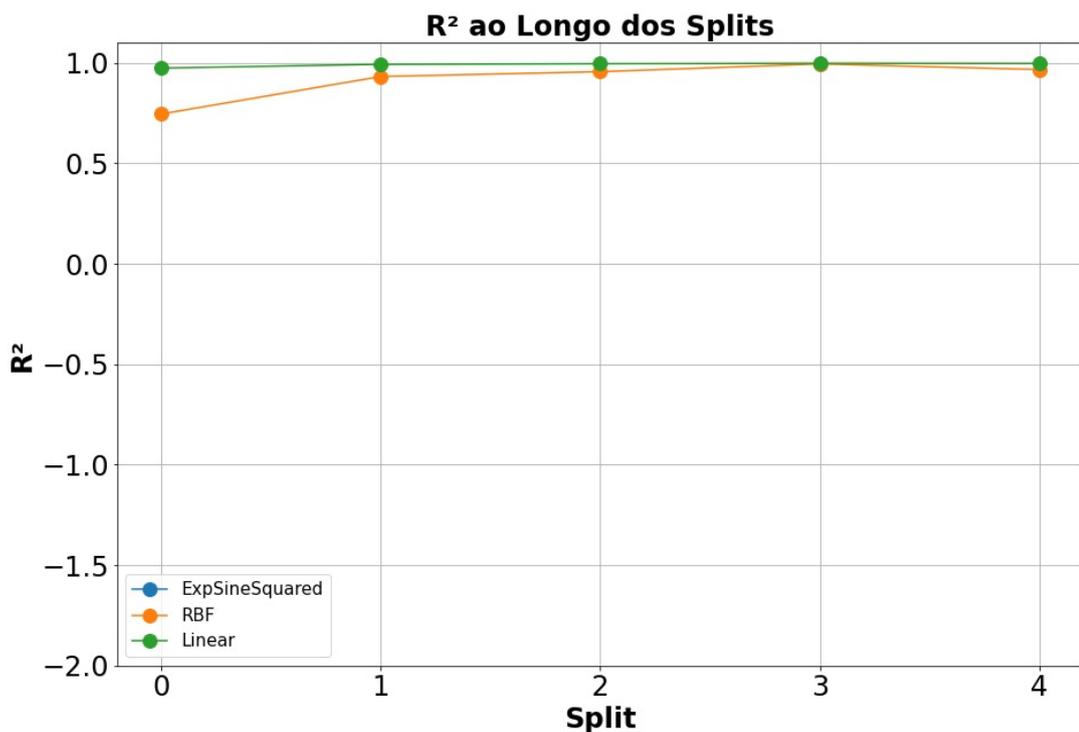


Fonte: Autoria própria, 2021

Dessa vez, podemos perceber qualitativamente que o *kernel* linear obteve um desempenho bastante superior aos demais. O *kernel* periódico foi incapaz de realizar o ajuste aos dados de tendência e o *kernel* SE, embora tenha conseguido realizar um ajuste aos dados de treino, não foi adequado para os dados de teste indicando, novamente, um *overfit* do modelo.

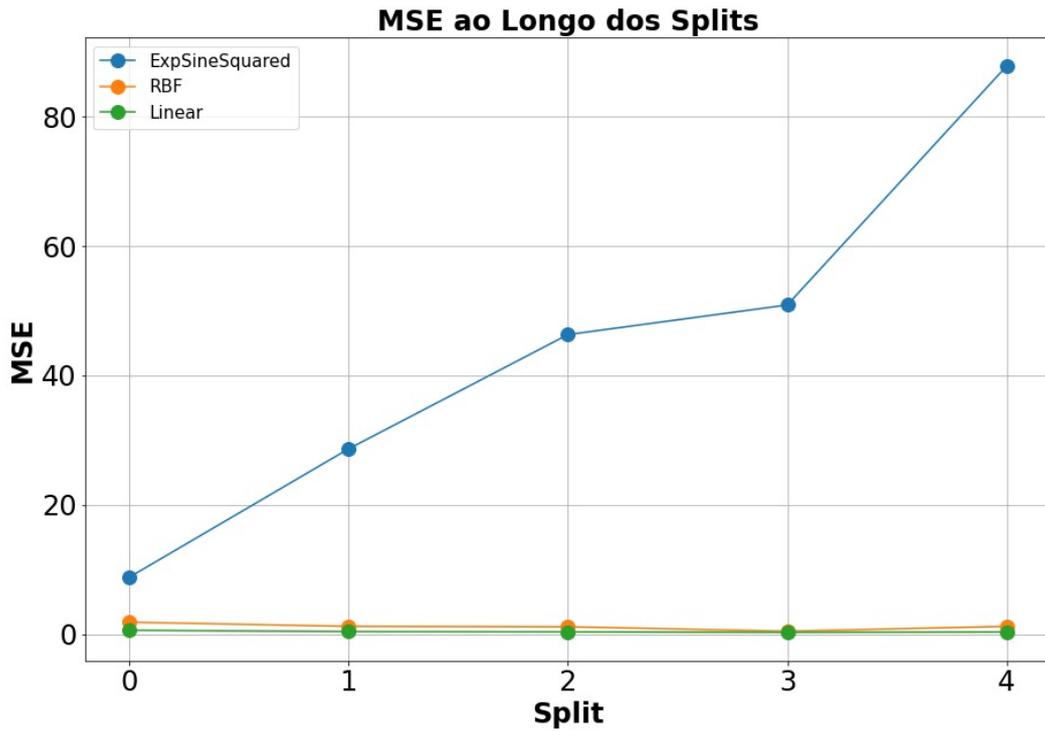
Uma comparação quantitativa das métricas propostas para esses 3 *kernels* ao longo dos *splits* da validação cruzada pode ser observada nas Figuras 31, 32 e 33. Podemos verificar uma superioridade nos resultados do *kernel* linear. No gráfico da Figura 31 vemos que o valor do r^2 do modelo de tendência se mantém próximo a 1 desde os splits iniciais. O r^2 para o modelo de sazonalidade apresentou resultados negativos e não são possíveis de visualizar de acordo com a escala do gráfico. Nas outras métricas de avaliação MSE e MAPE, podemos perceber que os *kernels* SE e linear apresentaram resultados bastante similares

Figura 37 – Comparação do R^2 ao longo dos splits - apenas tendência



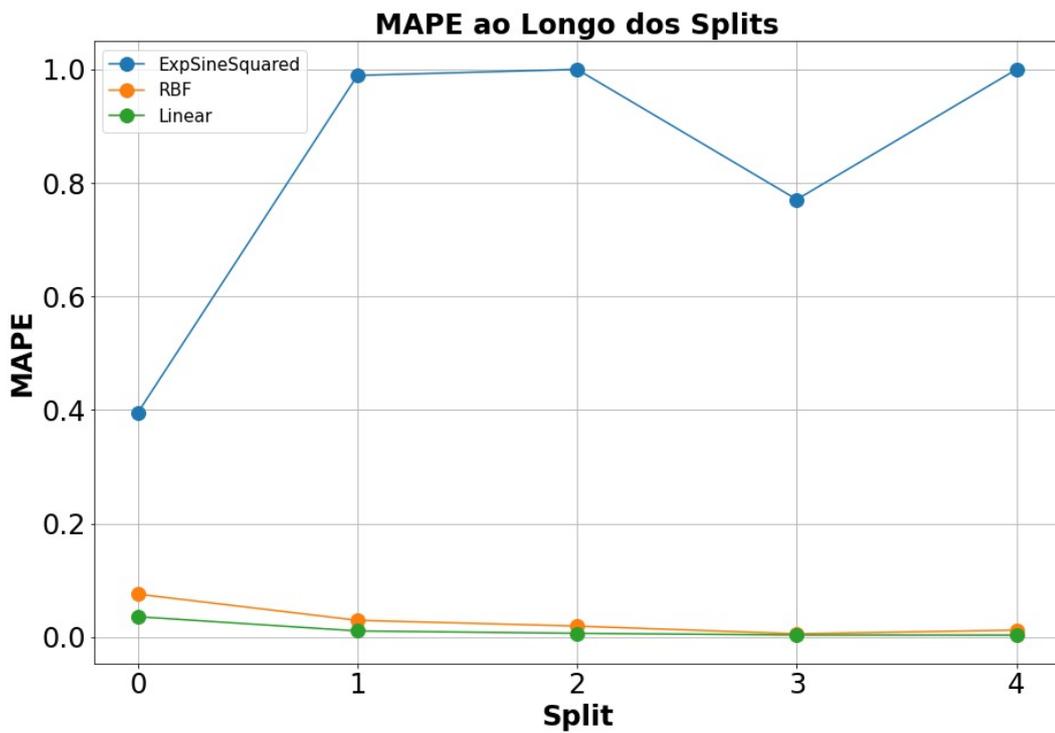
Fonte: Autoria própria, 2021

Figura 38 – Comparação do MSE ao longo dos splits - apenas tendência



Fonte: Autoria própria, 2021

Figura 39 – Comparação do MAPE ao longo dos splits - apenas tendência

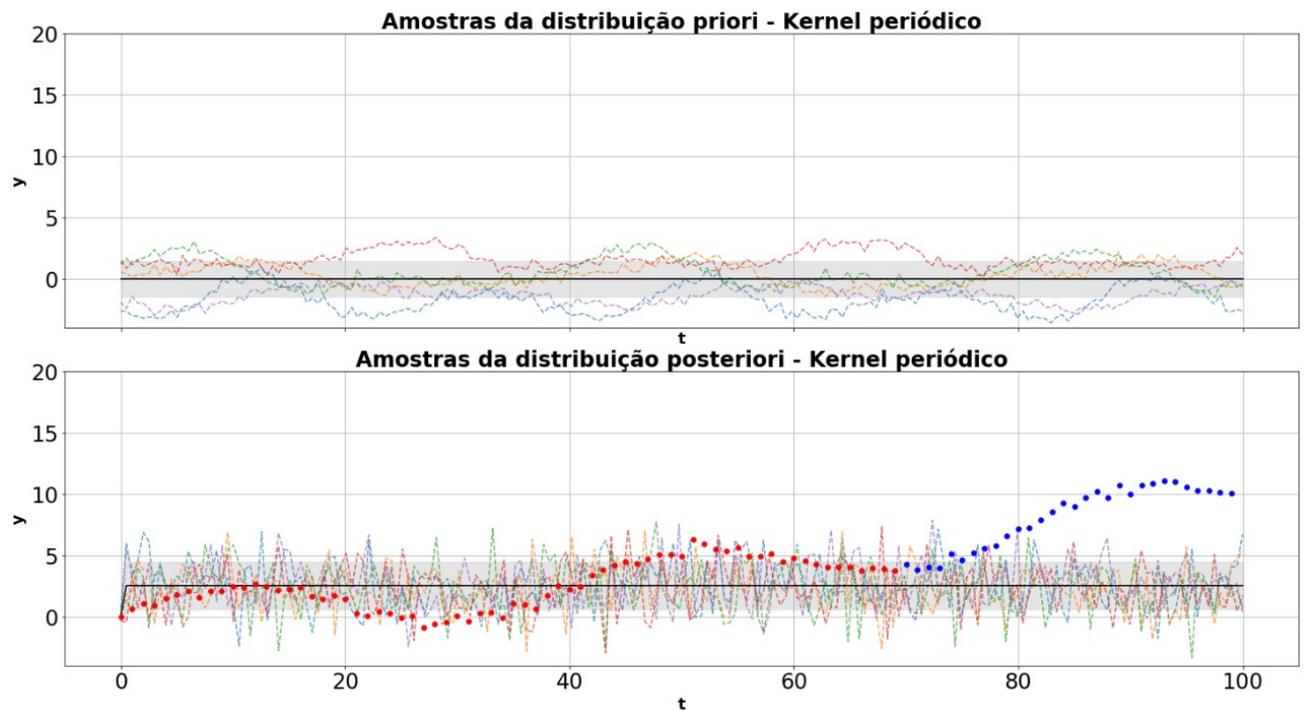


Fonte: Autoria própria, 2021

4.3 Série temporal com componente de tendência e de sazonalidade

Para o terceiro problema base proposto, podemos observar nas Figuras 40, 41, 42, 43 amostras das distribuições prioris e posterioris para os *kernels* discutidos na seção 3.3.2.

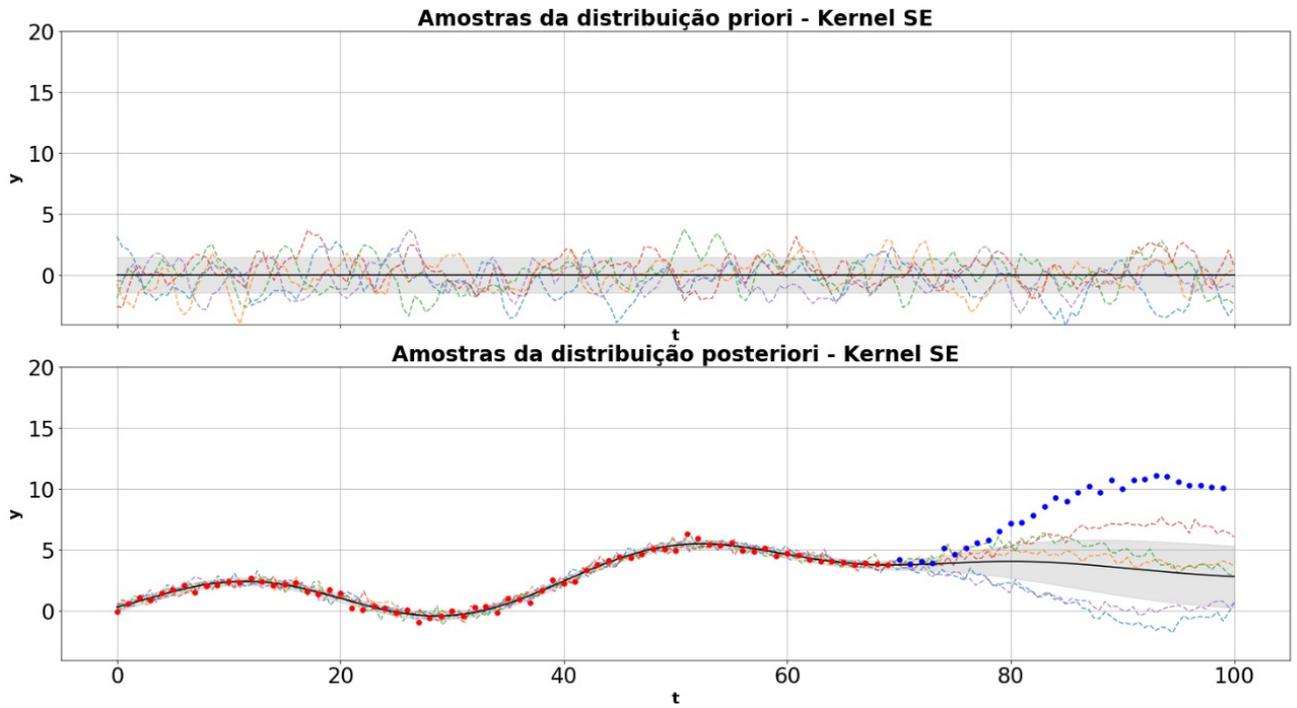
Figura 40 – Amostras a priori e a posteriori do *kernel* periódico para a série com tendência e sazonalidade



Fonte: Autoria própria, 2021

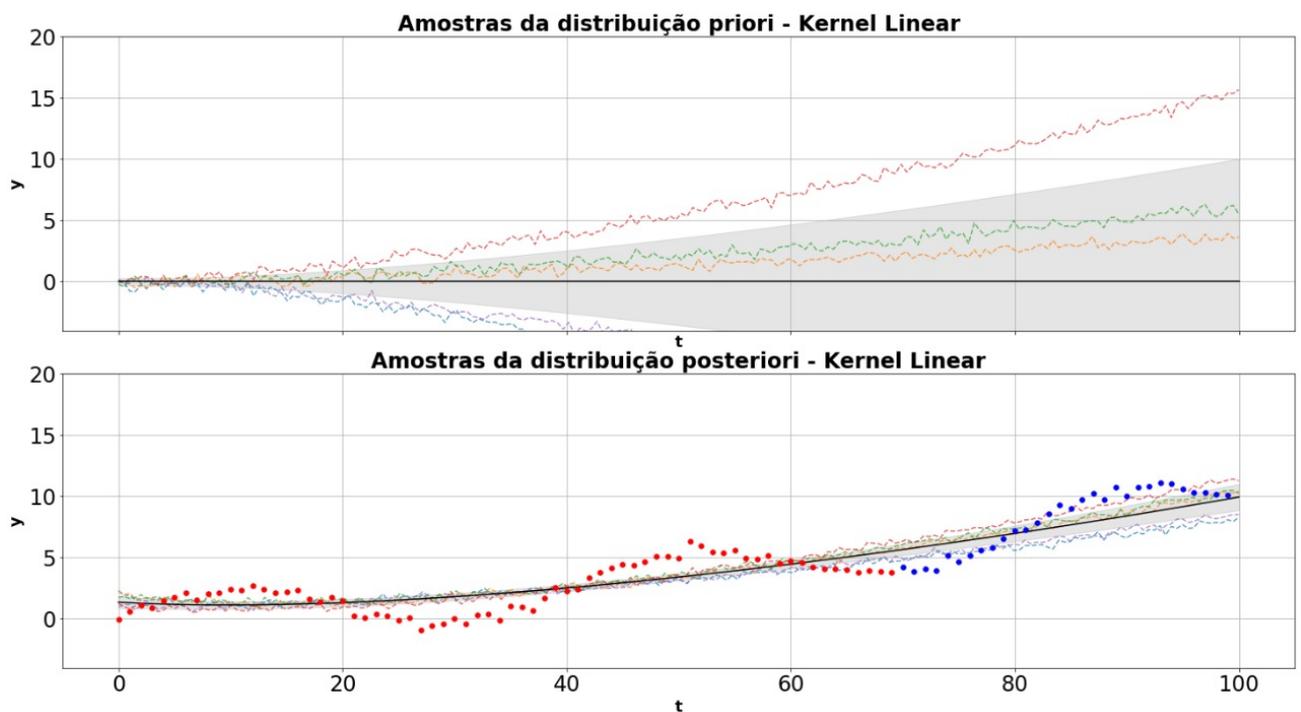
- O *kernel* puramente periódico não foi capaz de se ajustar bem aos dados, já que não consegue capturar a componente de tendência
- O *kernel* SE, embora seja capaz de realizar um ajuste aos dados de treino, logo começa a realizar previsões apenas na média de fenômeno. O conjunto de teste apresentou um resultado muito ruim para o r^2
- O *kernel* linear foi capaz de acompanhar apenas a tendência do fenômeno, assim como um modelo de média móvel
- O *kernel* mais adequado para o problema, conforme esperado, é uma soma de um *kernel* linear com um *kernel* periódico, já que leva em consideração tanto a tendência quanto a sazonalidade da série temporal em questão

Figura 41 – Amostras a priori e a posteriori do *kernel* SE para a série com tendência e sazonalidade



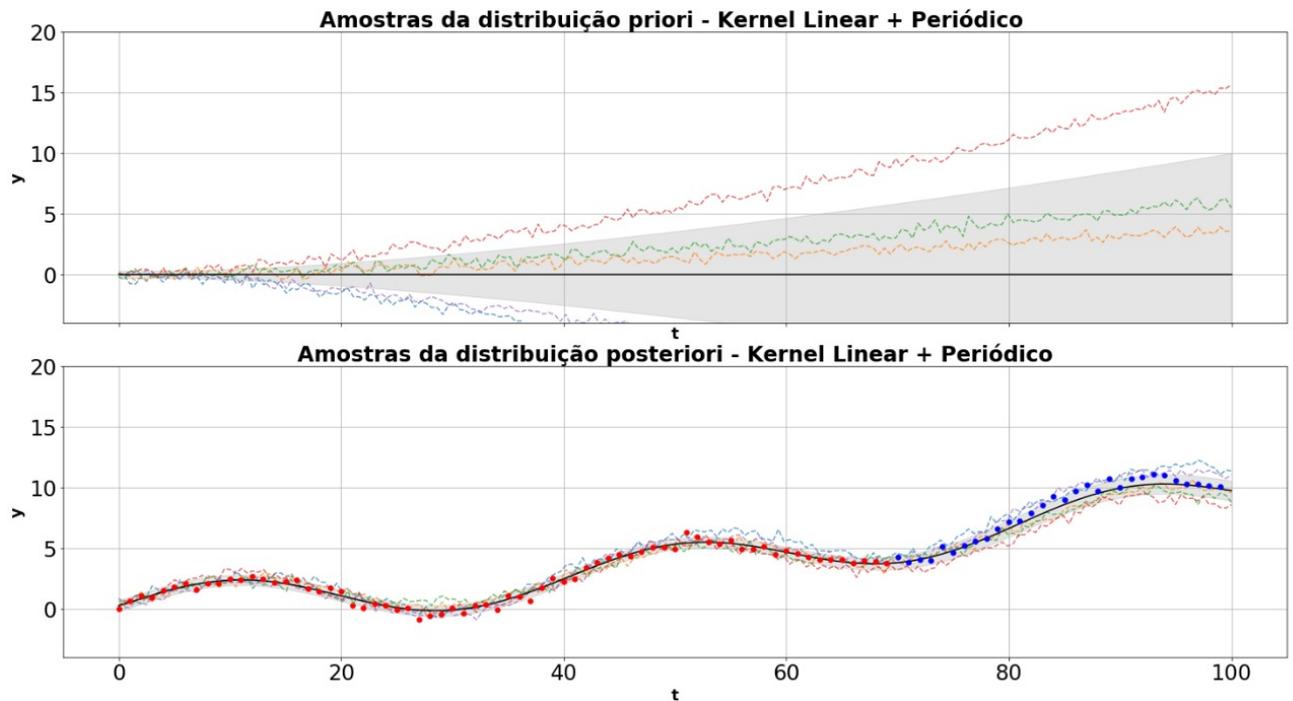
Fonte: Autoria própria, 2021

Figura 42 – Amostras a priori e a posteriori do *kernel* linear para a série com tendência e sazonalidade



Fonte: Autoria própria, 2021

Figura 43 – Amostras a priori e a posteriori do *kernel* linear + periódico para a série com tendência e sazonalidade



Fonte: Autoria própria, 2021

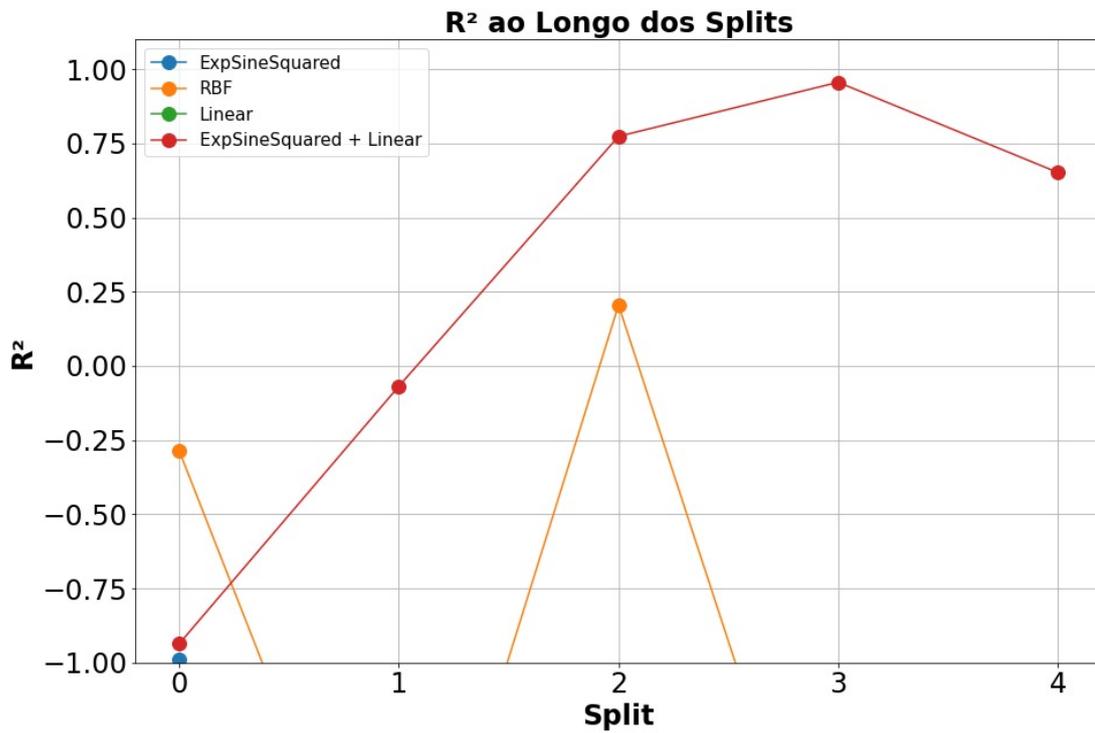
Uma comparação quantitativa das métricas propostas para esses 3 *kernels* ao longo dos *splits* da validação cruzada pode ser observada nas Figuras 44, 45 e 46. Podemos verificar que, para o modelo mais adequado, ainda foi necessário uma certa quantidade de dados para que o ajuste se dê de forma adequada, já que nos *splits* iniciais, o resultado de explicabilidade (r^2) foi muito baixo. Os demais modelos possuem um r^2 muito ruim. Nas demais métricas, também vê-se uma superioridade do *kernel* soma.

4.4 Previsão do acumulado de casos de Covid

Para a busca em grade proposta na seção 3.4.2, os seguintes *kernels* e parâmetros obtiveram os melhores resultados para a estrutura de regressão polinomial e regressão semi-paramétrica, respectivamente:

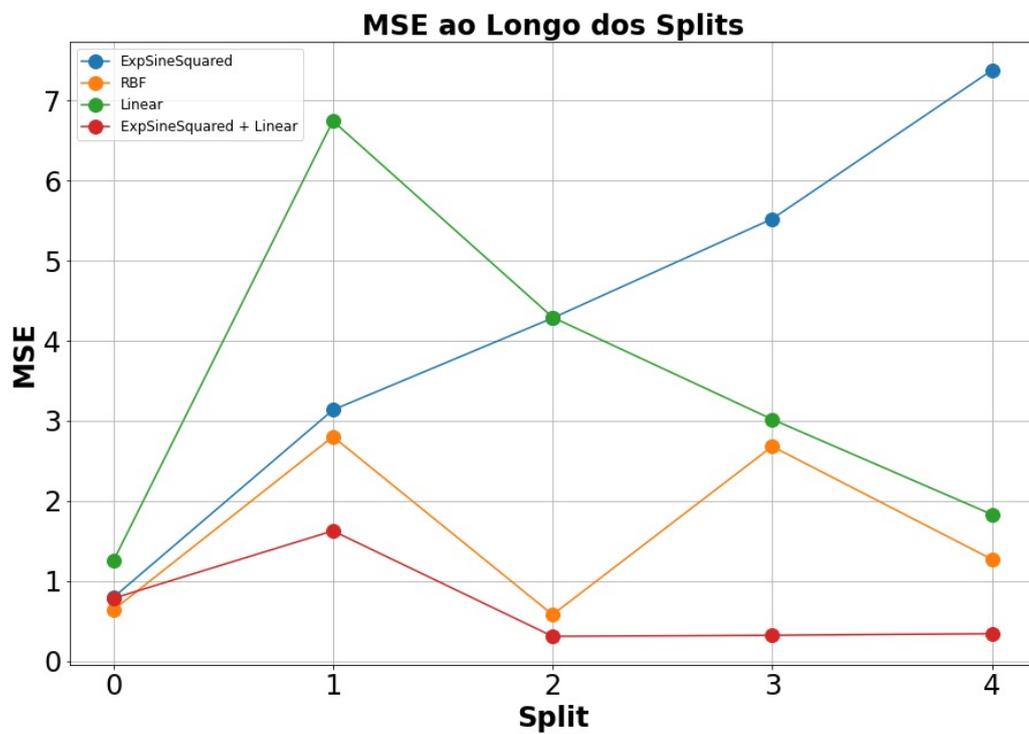
- *kernel* constante com valor de nível igual a 2 somado ao *kernel* linear com grau 1 somado ao *kernel* de ruído com variância igual a 1.
- *kernel* SE com *lengthscale* igual a 1 somado ao *kernel* linear com grau 1 somado ao *kernel* de ruído com variância igual a 1.

Figura 44 – Comparação do R^2 ao longo dos splits - tendência e sazonalidade



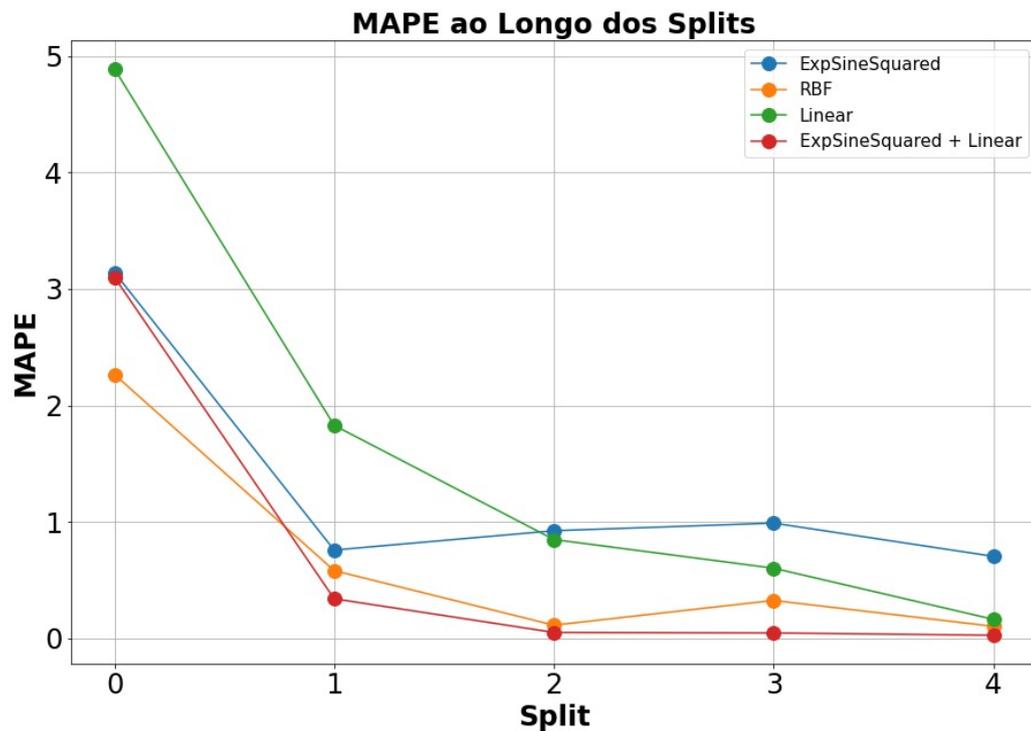
Fonte: Autoria própria, 2021

Figura 45 – Comparação do MSE ao longo dos splits - tendência e sazonalidade



Fonte: Autoria própria, 2021

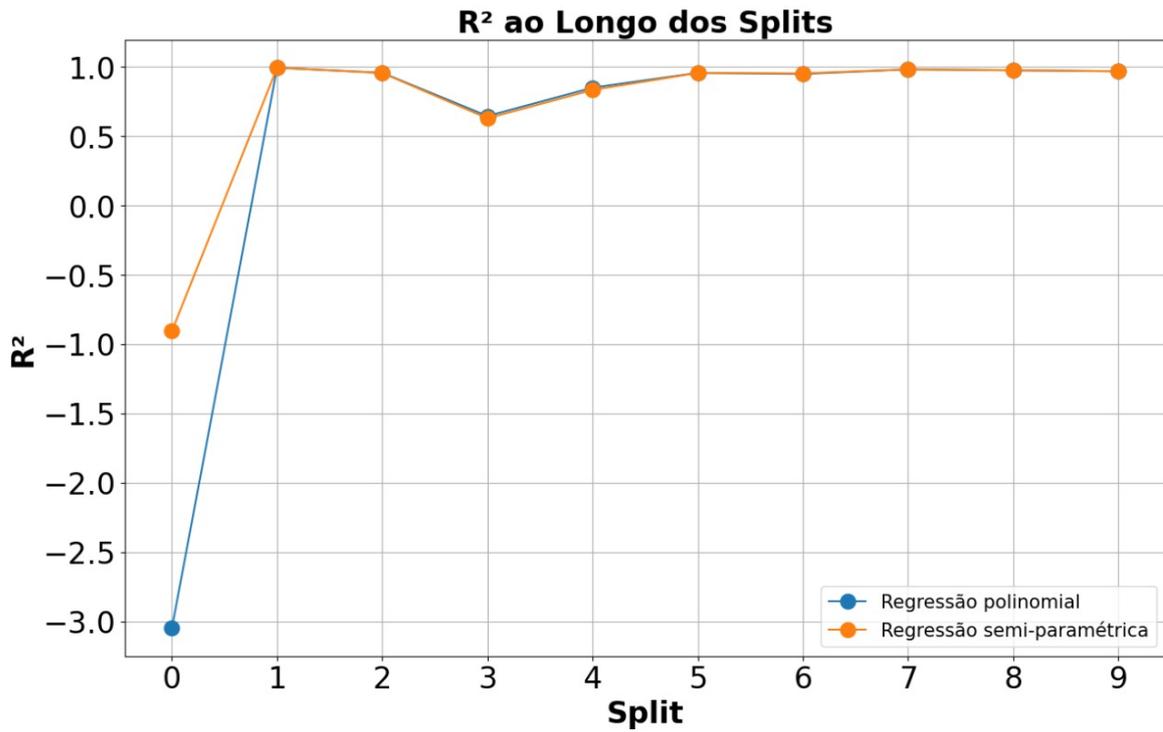
Figura 46 – Comparação do MAPE ao longo dos splits - tendência e sazonalidade



Fonte: Autoria própria, 2021

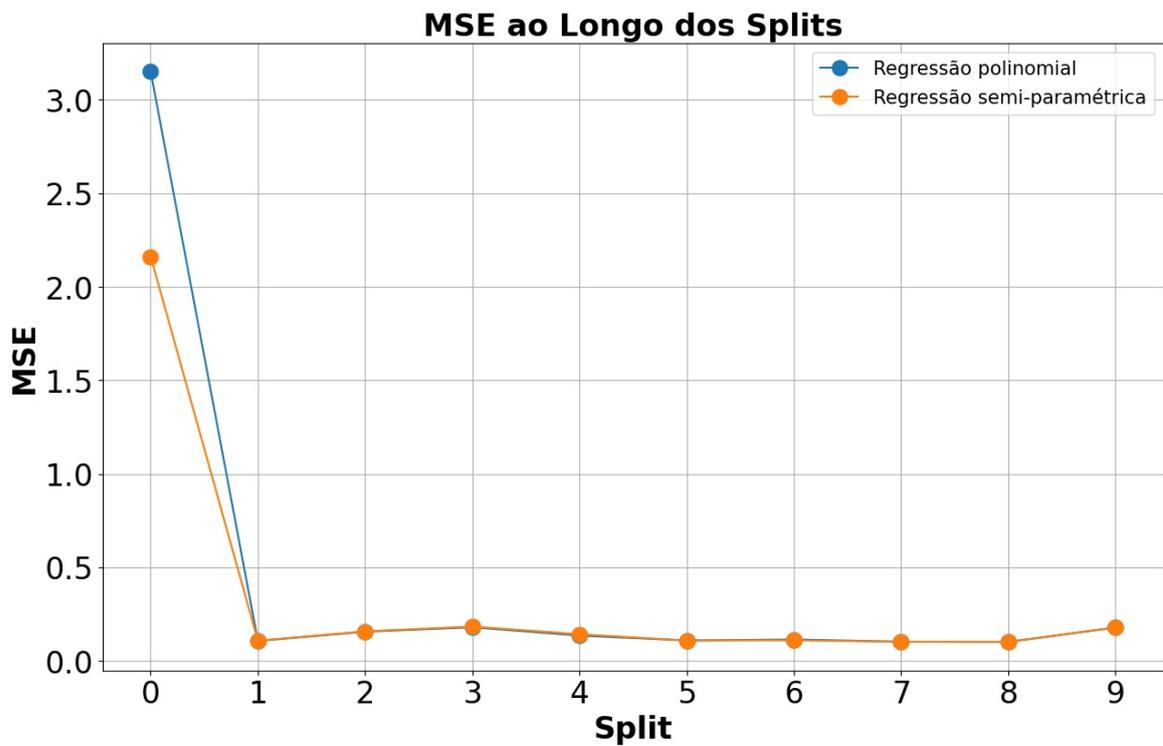
Ambas as estruturas apresentaram resultados muito bons nas métricas de avaliação desde os splits iniciais, conforme pode ser observado nas Figuras 47, 48 e 49. A partir dos gráficos, podemos afirmar que, dado o problema do acumulado de casos, as estruturas de *kernel* para regressão polinomial e para a regressão semi-paramétrica demonstram uma necessidade de poucos dados para a realização, já que no split 1, que utiliza apenas 20% dos dados, o r^2 já possui um valor muito próximo a 1. O MAPE e o MSE, que indicam a distância entre as saídas previstas e os dados reais também já apresentam um resultado muito próximo de 0 desde o split 1.

Figura 47 – Comparação do R^2 ao longo dos splits - acumulado de casos



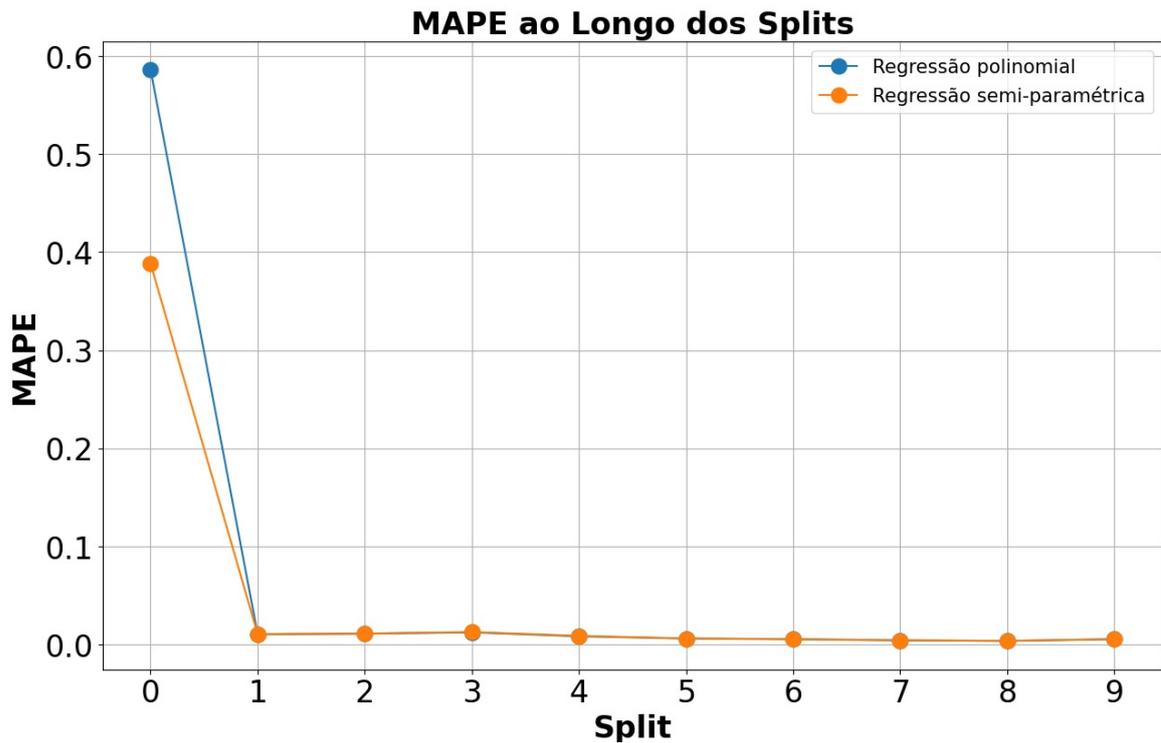
Fonte: Autoria própria, 2021

Figura 48 – Comparação do MSE ao longo dos splits - acumulado de casos



Fonte: Autoria própria, 2021

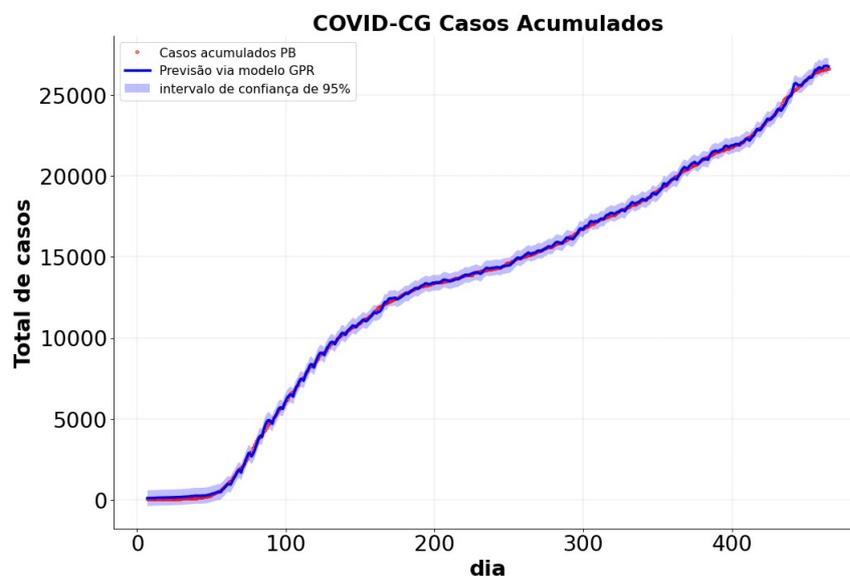
Figura 49 – Comparação do MAPE ao longo dos splits - acumulado de casos



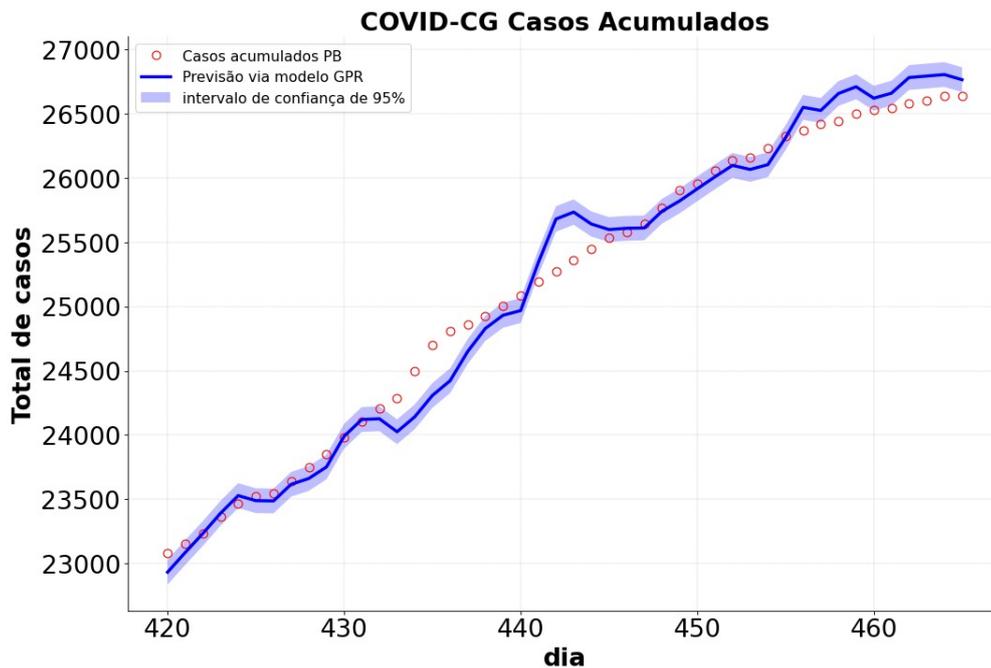
Fonte: Autoria própria, 2021

Nas Figuras 50 e 51, podemos verificar os dados previstos com o modelo de regressão semi-paramétrica. Na Figura 50 analisamos todo o intervalo de dados, incluindo *splits* de treinamento. Na Figura 51 mostramos a previsão apenas nos 10% de dados do último *split*, já que o modelo não utilizou esses dados em nenhuma etapa do treinamento.

Figura 50 – Predições e dados reais para o todo o conjunto de dados de casos acumulados

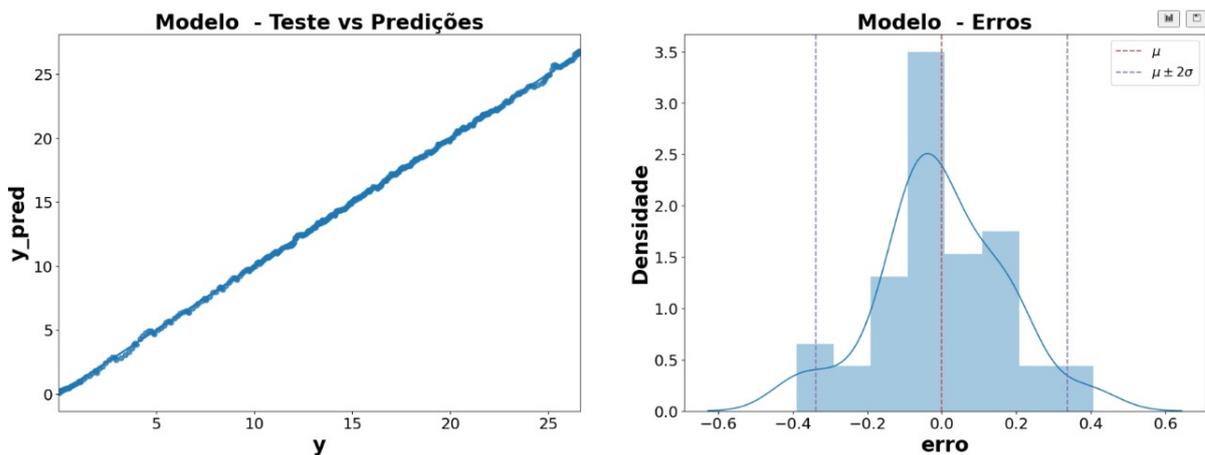


Fonte: Autoria própria, 2021

Figura 51 – Predições e dados reais para o último *split* de teste

Fonte: Autoria própria, 2021

Por fim, na Figura 52, podemos observar os erros cometidos pelo modelo no último *split* de teste. Podemos notar que os erros possuem uma distribuição homogênea centrada na média 0.

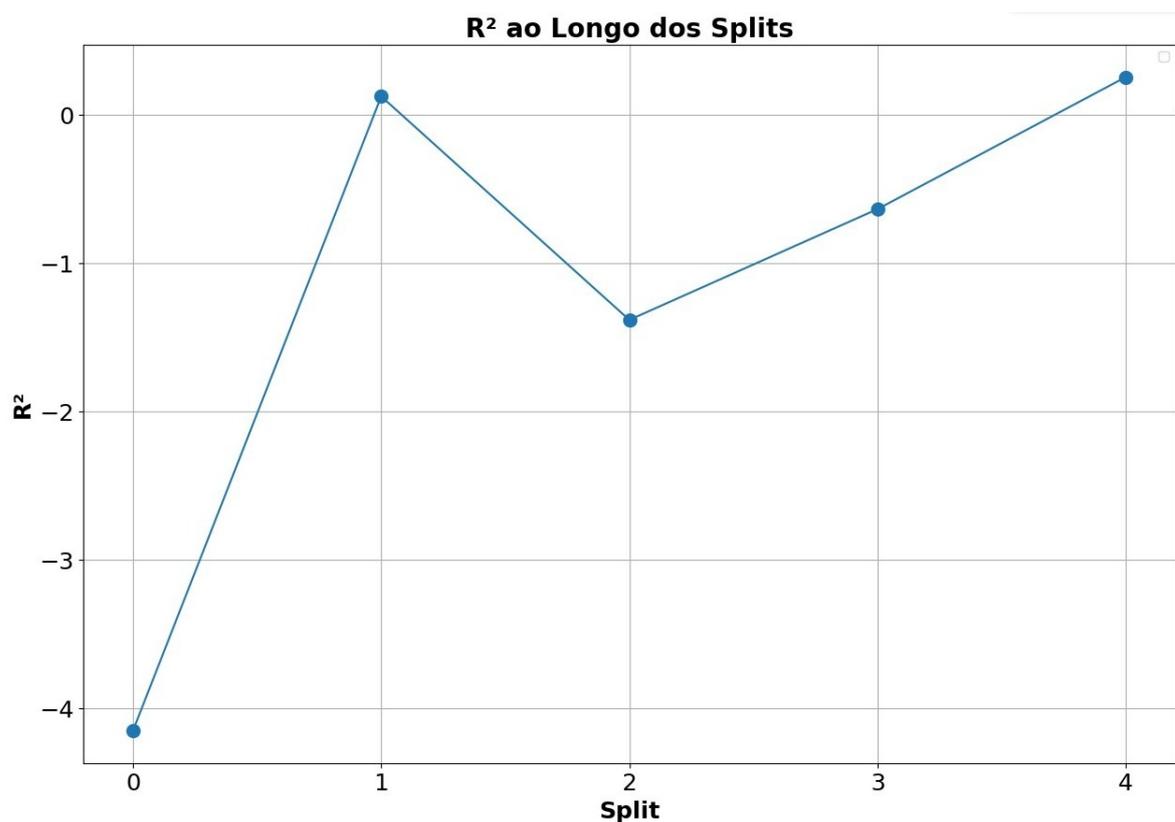
Figura 52 – Erros do modelo para o último *split* de teste

Fonte: Adaptação, (ORDUZ, 2019)

4.5 Previsão da média móvel de casos de Covid

Para a modelagem do problema e a estrutura de *kernels* proposta na seção 3.5 deste documento, obtivemos os seguintes resultados:

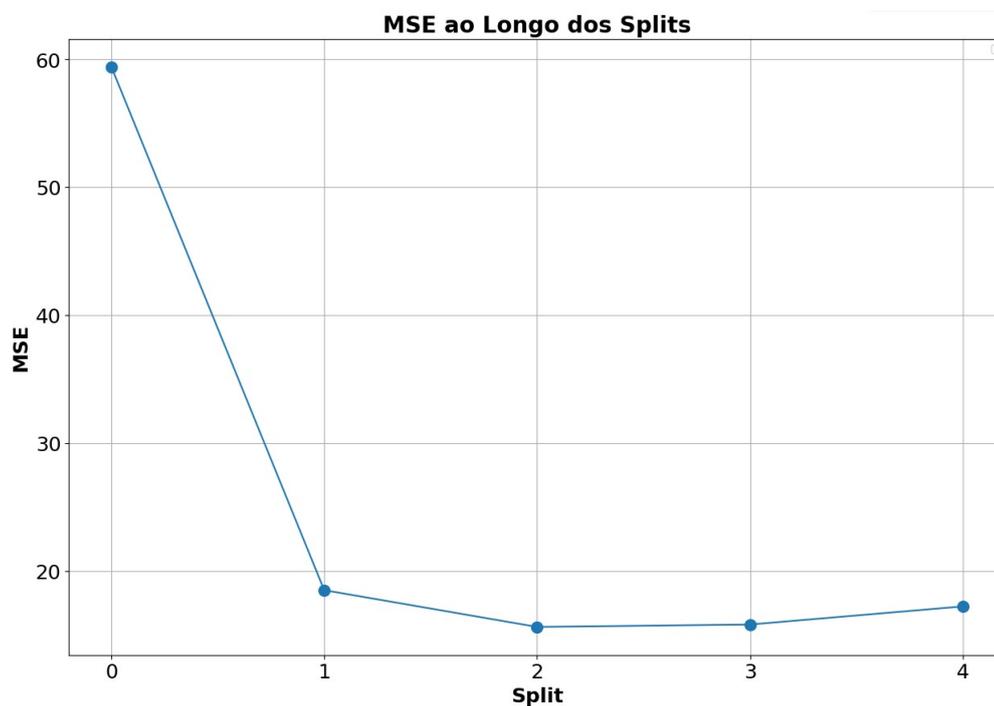
- Para a estrutura de *kernels* *Trend*, *cyclical*, *irregular*, houve um problema de convergência do modelo. Isso se deu porque a matriz de correlação construída a partir da função *kernel* não ficou da forma semipositiva definida.
- Para a estrutura de *kernels* de regressão semi-paramétrica, os resultados do r^2 foram em muitos splits, negativos. Isso significa que o modelo não foi capaz de explicar os dados de média móvel de casos diários. As Figuras a seguir mostram os resultados para o r^2 , MSE e MAPE para o modelo que apresentou um melhor ajuste.

Figura 53 – Comparação do R^2 ao longo dos splits - média móvel de casos

Fonte: Autoria própria, 2021

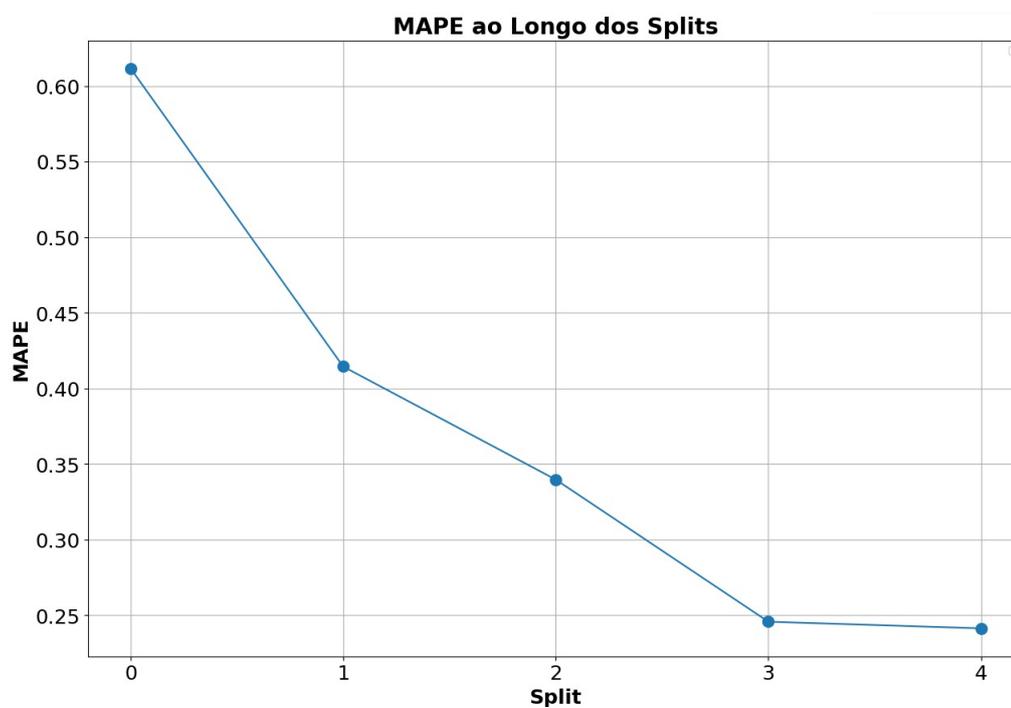
Conforme podemos observar, houve uma melhora nas métricas de avaliação com o aumento do número de dados de treino. No split 4, em que usamos a maior quantidade de dados, o erro médio percentual foi de pouco menos de 25%, ou seja, o valor previsto para a média móvel de casos diários é aproximadamente 25% maior ou menor do que o real. Na Figura 56, podemos visualizar as previsões do modelo para todo o intervalo de treinamento. Na Figura 57, comparamos as previsões e os dados reais apenas para o último *split* de teste, ao qual o modelo não teve acesso durante a etapa de treino.

Figura 54 – Comparação do MSE ao longo dos splits - média móvel de casos



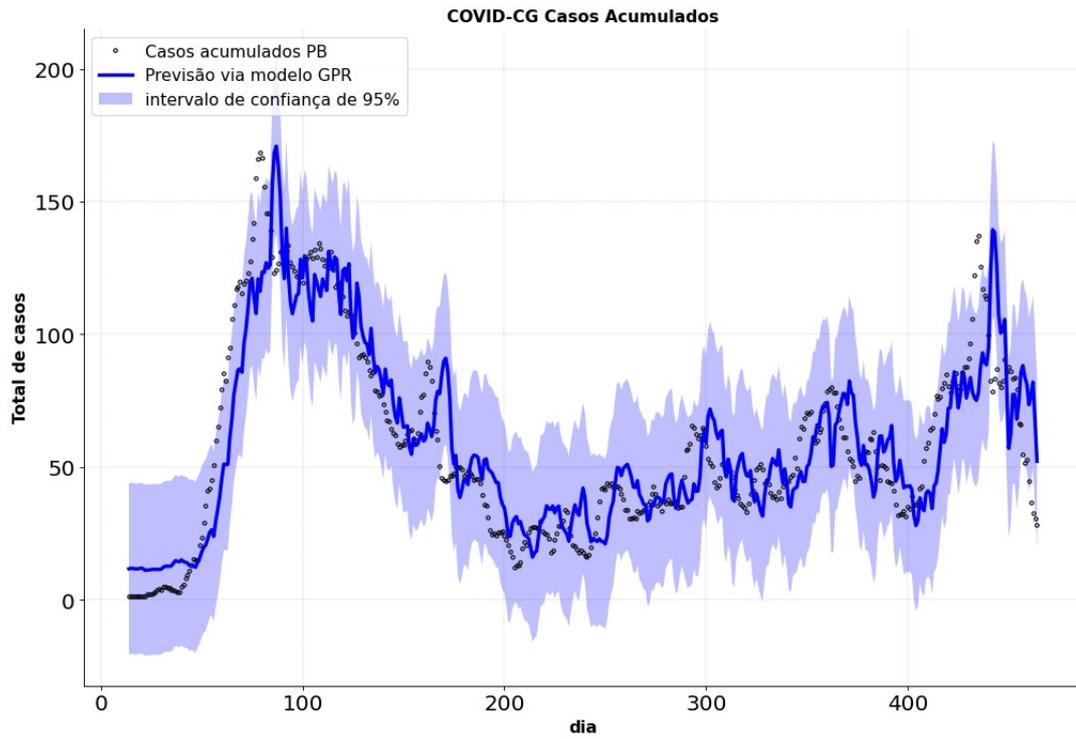
Fonte: Autoria própria, 2021

Figura 55 – Comparação do MAPE ao longo dos splits - média móvel de casos



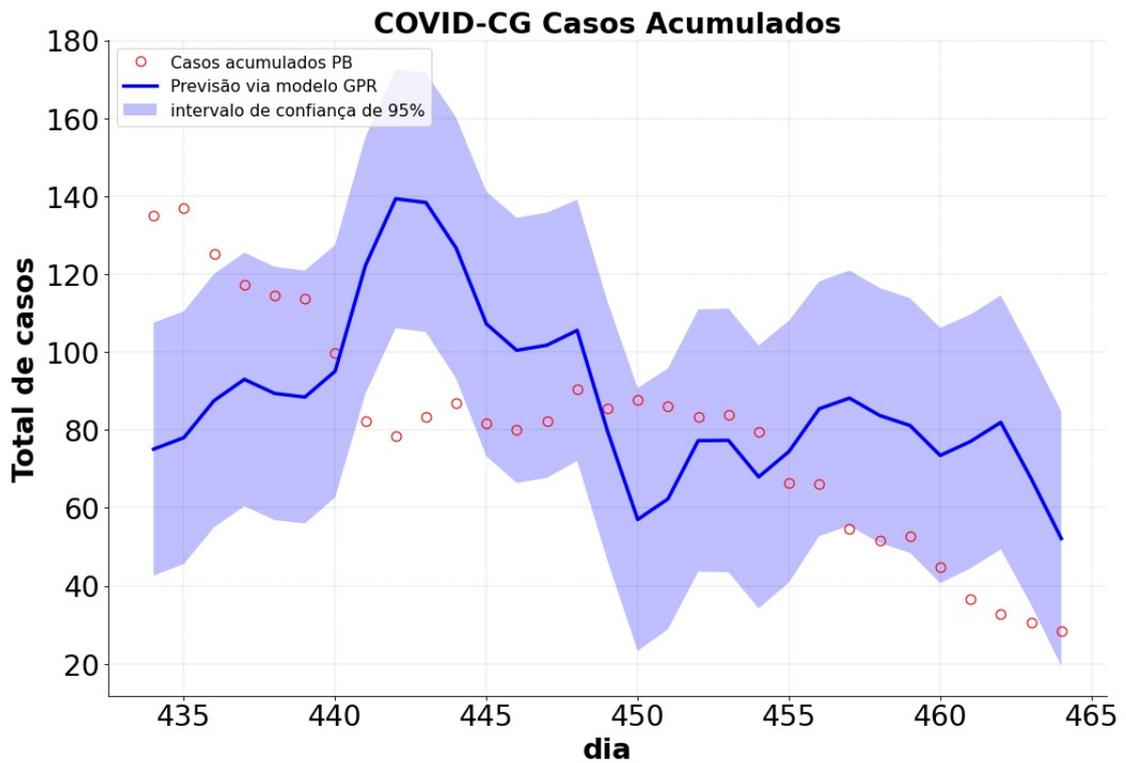
Fonte: Autoria própria, 2021

Figura 56 – Predições e dados reais para o todo o conjunto de dados de média móvel



Fonte: Autoria própria, 2021

Figura 57 – Predições e dados reais para o último *split* de teste de média móvel



Fonte: Autoria própria, 2021

Para esse problema, pode-se perceber que o ajuste não se deu de forma adequada. Na Figura 57 vemos que há um grande intervalo de confiança para o período, isso é, o modelo não tem assertividade quanto às predições. Dessa maneira, é preciso destacar que é necessário que haja uma continuação na busca de *kernels* para construção de um modelo adequado. Técnicas mais avançadas de combinação de kernels, conforme propostas por (DUVENAUD, 2014) podem ser aplicadas para a construção de novas hipóteses para espaços de busca.

5 Conclusão

Neste trabalho de conclusão de curso descreveu-se os princípios de funcionamento da Regressão por Processo Gaussiano, com destaque para a comparação entre a abordagem frequentista e a abordagem Bayesiana de aprendizagem. Além disso, apresentou-se uma fundamentação acerca da previsão de séries temporais, foco da aplicação do trabalho.

Foram propostos 3 problemas básicos para o estudo das características das funções *kernel* do GPR e para o estudo do processo de aprendizagem em si. A partir destes, pudemos visualizar amostras das funções de distribuição a priori e a posteriori do modelo e perceber que tipos de estruturas de *kernels* podem se adequar a cada problema específico.

Esses problemas embasaram e motivaram uma aplicação prática do GPR para a previsão da evolução de casos de infecção da Covid-19 na cidade de Campina Grande: tanto para o acumulado de casos quanto para a média móvel de novos casos. Pudemos propor hipóteses acerca de que funções *kernels* seriam mais adequadas para cada tipo de problema a partir da análise das séries temporais. Os resultados obtidos para as previsões foram mostrados e discutidos na seção 4 deste trabalho.

Pudemos perceber que o processo de treinamento para o problema da previsão de casos acumulados se deu de forma adequada e rápida, já que apenas uma pequena parcela dos dados totais precisou ser usada para obtermos um modelo com alto nível de explicabilidade (grande r^2). Isso mostra que com uma estrutura adequada de *kernel* podemos esperar um bom desempenho desse algoritmo.

Por outro lado, para o problema de previsão da média móvel de casos diários, que possui uma dificuldade muito maior, não foi possível encontrar um modelo com um desempenho satisfatório dentro do espaço de busca considerado. Esse espaço de busca foi limitado à duas estruturas de *kernels* proposta na seção 2.2.2 — a estrutura de regressão semi-paramétrica e a estrutura *Trend, cyclical, irregular*. A média móvel de casos diários possui uma curva complexa, influenciada por diversos fatores sociais. A utilização de variáveis autoregressoras, conforme aplicada neste trabalho, é uma grande simplificação utilizada para a tarefa de previsão. Devido aos benefícios sociais que um bom modelo de previsão da evolução da doença pode trazer, bem como à grande quantidade de estruturas de busca que ainda podem ser exploradas no GPR, aponta-se como trabalho futuro a expansão do conjunto de *kernels* considerados para a tarefa de regressão, assim como o aumento ou mudança das *features* utilizadas.

O principal desafio para o discente na execução deste trabalho foi a incorporação dos conceitos básicos necessários para o entendimento do algoritmo do GPR. Os tópicos envolvidos na Regressão por Processo Gaussiano são bastante pautados na teoria de

probabilidades e no ramo da estatística. Conhecimentos trabalhados nas disciplinas de base do curso de Engenharia Elétrica, tais como o Cálculo I, Álgebra Linear, Probabilidade e Estatística e Processos Estocásticos foram essenciais para o desenvolvimento do trabalho. Por fim, deve-se destacar que este o estudo do GPR permitiu o aprendizado de conceitos, ferramentas e habilidades complementares a formação curricular.

Referências

- ABDESSALEM, A. B. et al. Automatic kernel selection for gaussian processes regression with approximate bayesian computation and sequential monte carlo. *Frontiers in Built Environment*, v. 3, p. 52, 2017. ISSN 2297-3362. Disponível em: <<https://www.frontiersin.org/article/10.3389/fbuil.2017.00052>>. Citado na página 23.
- AILEEN, N. *Practical Time Series Analysis*. [S.l.]: O'Reilly, 2020. Citado 2 vezes nas páginas 34 e 35.
- BOX, G. E. et al. *Time Series Analysis, Forecasting and Control*. [S.l.]: John Wiley Sons, Inc, 2016. Citado na página 31.
- BROWNLEE, J. *What Is Time Series Forecasting?* 2020. Disponível em: <<https://machinelearningmastery.com/time-series-forecasting/>>. Citado na página 32.
- CAUSEVIC, S. *Frequentist vs. Bayesian Approaches in Machine Learning*. 2020. Disponível em: <<https://towardsdatascience.com/frequentist-vs-bayesian-approaches-in-machine-learning-86ece21e820e>>. Citado na página 16.
- DUVENAUD, D. K. *Automatic Model Construction with Gaussian Processes*. Tese (Doutorado) — University of Cambridge, 2014. Citado 7 vezes nas páginas 27, 28, 29, 30, 31, 50 e 72.
- FREITAS, N. de. *Machine Learning - Bayesian Learning*. 2013. Disponível em: <<https://www.youtube.com/watch?v=Fae0j1WN1zA&list=PLE6Wd9FR--EdyJ5lbF18UuGjcvVw66F6>>. Citado 3 vezes nas páginas 14, 18 e 26.
- GALIT, S.; KENNETH, L. *Practical Time Series Forecasting with R*. [S.l.]: Axelrod Schnall, 2016. Citado 2 vezes nas páginas 32 e 33.
- JAN, H. M.; GUILLAUME, L. *Illustration of prior and posterior Gaussian process for different kernels*. 2021. Disponível em: <https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_prior_posterior.html>. Citado 2 vezes nas páginas 25 e 29.
- ORDUZ, J. C. *Gaussian Processes for Time Series Forecasting (scikit-learn)*. 2019. Disponível em: <https://juanitorduz.github.io/gaussian_process_time_series/>. Citado 3 vezes nas páginas 14, 36 e 68.
- RASMUSSEN, C.; WILLIAMS, C. *Gaussian Processes for Machine Learning*. [S.l.]: MIT Press, 2006. Citado 5 vezes nas páginas 13, 19, 23, 24 e 26.
- RUSSEN, S. J.; NORVIG, P. *Artificial Intelligence, A Modern Approach*. [S.l.]: Alan Apt, 1995. Citado na página 16.
- SWASTANTO, B. A. *Gaussian Process Regression for Long-Term Time Series Forecasting*. Dissertação (Dissertação de Mestrado) — Delft University of Technology, Delft, 2016. Citado 3 vezes nas páginas 23, 24 e 26.