University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Forecasting Inflation in Brazil with Machine Learning Methods: Integrating Shrinkage Method for Variable Selection with Shapley Value Interpretation

Previsão de inflação no Brasil com métodos de aprendizado de máquina: integrando método de encolhimento para seleção de variáveis com interpretação por meio do valor de Shapley

Felipe Gonçalves Pereira

Dissertation submitted in partial fulfillment of the requirements of Universidade de Brasília for the degree of Master of Science in Applied Computing

Supervisor
Prof. Dr. João Gabriel de Moraes Souza

Brasília
2024

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

PP436f

Pereira, Felipe Gonçalves
Forecasting Inflation in Brazil with Machine Learning Methods: integrating shrinkage method for variable selection with shapley value interpretation / Felipe Gonçalves Pereira; orientador João Gabriel de Moraes Souza; co-orientador João Carlos Felix Souza. -- Brasília, 2024.
98 p.

Tese(Mestrado Profissional em Computação Aplicada) -- Universidade de Brasília, 2024.

1. Previsão de Inflação. 2. aprendizado de máquina. 3. Seleção de Variáveis. 4. Inteligência Artificial Explicável. 5. Shapley Value. I. Souza, João Gabriel de Moraes, orient. II. Souza, João Carlos Felix, co-orient. III. Título.

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Forecasting Inflation in Brazil with Machine Learning Methods: Integrating Shrinkage Method for Variable Selection with Shapley Value Interpretation

Previsão de inflação no Brasil com métodos de aprendizado de máquina: integrando método de encolhimento para seleção de variáveis com interpretação por meio do valor de Shapley

Felipe Gonçalves Pereira

Dissertation submitted in partial fulfillment of the requirements of Universidade de Brasília for the degree of Master of Science in Applied Computing

Prof. Dr. João Gabriel de Moraes Souza (Supervisor)
CIC/UnB

Prof. Dr. Peng Yaohao	Prof. Dr. Mathias Schneid Tessmann
University of Brasilia	IDP

Prof. Dr. Gladston Luiz da Silva
Postgraduate Program Coordinator in Applied Computing

Brasília, May , 2024

Dedication

To my beloved wife, Denise, for her unwavering support and endless love that lights up the darkest days of this journey; to my dear mother, Clarice, whose strength and wisdom have always guided me towards my aspirations; to my son, Davi Henrique, the joy and inspiration of my life, reminding me daily of the reasons I strive for excellence; and to my brothers, Adriano and André Tiago, for their constant encouragement and the bonds of brotherhood that have given me courage and comfort. This dissertation stands as a testament to the unyielding support and unconditional love you have all shared with me. It is to you, my family, I dedicate this work, with all my love and gratitude.

Acknowledgements

First of all, I would like to thank the blessings that God, in his eternal benevolence, has been pouring out on me. I am grateful for the privilege of my family and the unconditional support they give me in all the challenges that life presents me with.

My gratitude to my dear and beloved wife Denise, the most virtuous soul, the purest and wisest heart, a true gift from God; my son Davi Henrique; my mother Clarice. I thank them for their joint effort to provide every opportunity and support, without hesitation, throughout my life.

Secondly, I would like to thank my supervisor, Prof. Dr. João Gabriel, for his patience, wisdom, and continuous support throughout the efforts to produce this work. His leadership skills and balance in the course of his guidance, coupled with his vast knowledge, were fundamental to the development and achievements associated with this research. His mentorship was a true blessing.

Resumo

Esta dissertação busca identificar o modelo de apredizagem de máquina (ML) não linear mais eficaz na previsão do Índice de Preços ao Consumidor (IPCA) mensal usando um método de seleção prévio de variáveis baseado em modelo de encolhimento para escolher os preditores mais significantes. Além disso, o estudo visou analisar os resultados da previsão usando um método de inteligência artificial explicável (XAI) indepedente de modelo chamado Shapely Value, que pode fornecer informações sobre as previsões do modelo de ML não linear.

Foi utilizado um conjunto de dados abrangendo o período de agosto de 2010 a janeiro de 2024, com 156 preditores. A partir dessa base de dados, foi realizada a seleção dos preditores mais significativos através de um loop que aplica a eliminação recursiva de variáveis (Recursive Feature Elimination - RFE) utilizando o modelo ElasticNet em cada mês do período de treinamento. Ao todo, foram realizadas 156 execuções do algoritmo de RFE, isolando os 30 preditores mais frequentes, aplicados aos modelos não lineares de ML.

Os resultados das previsões evidenciaram o Gradient Boosting como o modelo mais eficaz, apresentando os melhores indicadores de acurácia e significância no teste de hipótese. A incorporação do Shapley Value aprimorou significativamente a interpretabilidade do modelo vencedor, oferencendo insights sobre as contribuições individuais de variáveis e mitigando a natureza de "caixa preta" dos modelos de ML. Os resultados evidenciaram a importância dos proxies para a variável-alvo nas previsões com contribuições significativas quando comparados com outros indicadores econômicos utilizados.

Palavras-chave: Previsão de Inflação, aprendizado de máquina, Seleção de Variáveis, Inteligência Artificial Explicável, Shapley Value

Abstract

This dissertation seeks to identify the most effective non-linear machine learning (ML) model for forecasting the monthly Brazilian Consumer Price Index (IPCA). It employs a prior feature selection (variable selection) method based on a shrinkage model to choose the most significant predictors. Additionally, the study aims to analyze prediction results using a model-agnostic explainable artificial intelligence (XAI) method called Shapley Value, which provides insights into non-linear model predictions.

A dataset covering the period from August 2010 to January 2024 was utilized, containing 156 predictors. From this database, the most significant predictors were selected through a recursive feature elimination (RFE) process using the ElasticNet model for each month of the training period. In total, 156 executions of the RFE algorithm were performed, isolating the 30 most frequent predictors to be applied to non-linear ML models.

The prediction results indicated that Gradient Boosting was the most effective model, demonstrating the best accuracy and significance indicators in hypothesis testing. The incorporation of Shapley Value significantly enhanced the interpretability of the winning model, providing insights into the contributions of individual variables and mitigating the “black box” nature of ML models. The results highlighted the importance of proxies for the target variable in predictions, with significant contributions compared to other economic indicators used.

Keywords: Inflation Forecasting, Machine Learning, Variable Selection, Explainable Artificial Intelligence (XAI), Shapley Value

Contents

List of Figures

List of Tables

Acronyms

BCB Brazilian Central Bank.

BIC Bayesian information criterion.

CPI Consumer Price Index.

EDA Exploratory Data Analysis.

ETL Extract, Transform, and Load.

I.I.D. Independently and Identically Distributed.

IBGE Instituto Brasileiro de Geografia e Estatística.

IMF International Monetary Fund.

IPCA Índice Nacional de Preços ao Consumidor Amplo.

LASSO Least Absolute Shrinkage and Selection Operator.

MAE Mean Absolute Error.

MCS Model Confidence Set.

ML Machine Learning.

MSE Mean Squared Error.

OLS Ordinary Least Squares.

RFE Recursive Feature Elimination.

RMSE Root Mean Squared Error.

RSS Residual Sum of Squares.

SHAP Shapley Additive Explanation.

STL Seasonal-Trend decomposition using Loess.

WoS Web of Science.

XAI Explainable Artificial Intelligence.

Chapter 1

Introduction

1.1 Contextualization

Investing financial resources, whether for non-financial corporations or financial institutions like banks, inherently carries considerable risk. The Treasury Department is pivotal in managing financial risks associated with investments across diverse corporate avenues, being crucial in the banking sector.

As highlighted by [Zaman et al., 2023], a financial institution's treasury oversees monetary assets, liabilities, and pertinent financial risks. This encompasses tasks ranging from liquidity management to managing exposure to fluctuating exchange and interest rates. Consequently, these functions encompass various market operations, including transactions in foreign currencies, securities, and derivatives.

Given the diverse products and currencies commercial banks manage, imbalances in their balance sheets are not uncommon. In these situations, the Treasury Department becomes instrumental in ensuring equilibrium by engaging in necessary market trades. Such operations, while essential, expose the institution to market risks. As elucidated by [CFA, 2023], market risk stems from potential shifts in the valuation of an organization's assets and liabilities, influenced by factors such as interest rate alterations, foreign exchange movements, stock price fluctuations, and other related elements. Thus, the onus falls upon the treasury department to monitor and anticipate market perturbations that could precipitate sizable losses.

Macroeconomic data releases, particularly those concerning inflation, stand at the forefront of determinants influencing market risk levels [Wang et al., 2023]. Unexpected surprises in these releases can drastically alter asset prices, especially in emerging markets like Brazil [Cakan et al., 2015]. This accentuates the role of macroeconomic variables in shaping investment strategies.

In the Brazilian context, inflation data holds immense weight when released. Fluctuations in this variable directly affect the term structure of interest rates, thereby influencing broader economic trends like aggregate spending and output [Lu & Wu, 2009]. Beyond the banking industry, inflation significantly influences the decision-making of various economic actors and individuals. For instance, in response to inflation spikes, they may seek higher interest rates to offset the anticipated devaluation of currency. Additionally, anticipated increases in inflation can prompt consumers to reduce discretionary spending, turning towards inflation-resistant investments such as real estate or gold. These behavioral changes can inadvertently slow economic growth, leading to ripple effects on employment and income levels. Brazil's volatile short-term inflationary trends further underscore the importance of accurate inflation forecasting for financial institutions [Garcia et al., 2017].

Brazil's embrace of inflation targeting in 1999, post the successful curtailment of hyperinflation via the Plano Real, underscores the significance of inflation in the nation's economic fabric [Garcia et al., 2017]. Effective inflation targeting mandates transparent and credible monetary policy execution. Thus, accurate forecasting of impending inflation trends becomes imperative to facilitate timely interest rate adjustments, essential for anchoring inflation expectations.

The crux of this study lies in addressing the challenge of inflation prediction. By leveraging machine learning techniques, this research seeks to empirically evaluate their forecasting efficacy, focusing on the Brazilian Broad Consumer Price Index (IPCA), as computed by the Brazilian Institute of Geography and Statistics (IBGE). Given the index's foundational role in contracts and its utilization by the Central Bank of Brazil (BCB) for inflation targeting, insights gleaned from this study stand to offer financial institutions a clearer understanding of impending inflation trajectories and, consequently, more robust mitigation strategies.

1.2 Problem

Accurate prediction of inflation is a longstanding imperative for both scholars and professionals. The significance of this endeavor traces back to foundational economic concepts like the real interest rate introduced by Fisher in 1930 [Fisher, Irving, 1930]. Predicting inflation is particularly challenging in emerging economies such as Brazil, where inflation rates tend to be higher and more volatile than in advanced economies. This volatility compresses the investment timeframe and amplifying the importance of short-term inflation predictions in emerging markets [Garcia et al., 2017].

Further complicating matters is the task of variable selection for inflation forecasting models. Given Brazil's relatively shorter span of macroeconomic time series coupled with the sheer volume of such variables, researchers often grapple with the Curse of Dimensionality [Verleysen & François, 2005]. Moreover, the high correlation among many macroeconomic indicators introduces additional complexity into the forecasting task [Aras, 2022]. Compounding these challenges is the inconsistent timing and frequency of the release of various Brazilian macroeconomic data series, which can impede timely forecasting.

However, the advent and continual evolution of computer technology offer promising avenues for addressing these challenges. Notably, Machine Learning (ML) algorithms, which have witnessed marked advancements in speed and accuracy, can efficiently process vast volumes of data (referred to as "Big Data" [1])—a feat beyond the capacity of traditional linear econometric models. Such data-rich explorations aim not only to predict outcomes but also to discern pivotal variables. This approach, dubbed hypothesis-generating analysis, stands in contrast to the hypothesis-testing research, where statistical techniques are deployed to fashion models grounded in pre-existing theories.

Despite the evident promise of ML in time series forecasting, its adoption in finance and macroeconomic contexts remains curtailed. Two principal obstacles underlie this hesitance. First, the often opaque nature of ML models poses interpretability challenges, leading some professionals in the economic and financial sectors to perceive a division between ML models and theory-driven econometrics [Iskhakov et al., 2020]. Second, real-world applications of ML frequently struggle with the intricate balance of bias and variance [Peng & Nagata, 2020].

The intersection of inflation forecasting, machine learning, and model interpretability constitutes a burgeoning arena of research. Financial institutions can wipe invaluable economic insights and enhance their predictive accuracy by synergizing extensive time series data with advanced ML and interpretability techniques. Notwithstanding these potential gains, a conspicuous lack of research specifically addressing inflation prediction in Brazil using these cutting-edge techniques exists. This research gap is the momentum for the present study, which aspires to predict the IPCA.

1.3 Research Questions

Highlighted in this text are the research questions aimed at addressing the problem.

1. Which nonlinear machine learning model demonstrates the most accurate forecast performance for the monthly Brazilian consumer price index (IPCA), utilizing variables (features) selected through a shrinkage method?

2. How can we make the predictions of our ML models interpretable?

[1] Denotes datasets with numerous variables and observations, often too extensive to be seamlessly integrated into traditional models.

1.4 Justification

The Brazilian banking sector faces mounting pressures to sustain profitability amid new competitors and a new set of regulatory mandates. Furthermore, the unprecedented repercussions of the Covid-19 pandemic have reverberated throughout the Brazilian economy and the global financial ecosystem.

Data from the International Monetary Fund (IMF) reveals a sharp ascent in global inflation rates post the Covid-19, surging from 2.9% in 2020 to 8.9% in 2022. This inflationary upswing is not isolated but is felt universally across all sectors. Several catalysts underpin this trend: disruptions in the supply chain, escalating commodity prices, and expansive monetary policies—all of which can trace their origins to the lockdowns initiated during the pandemic. The geopolitical tensions between Ukraine and Russia further compounded these challenges. The repercussions of this inflationary surge permeate businesses, inducing volatility and prompting enterprises to reevaluate their risk mitigation strategies [imf, 2023].

For financial institutions, the ramifications are profound. Spikes in inflation can trigger erratic price fluctuations, resulting in potential misallocation. Unexpected inflationary peaks contribute to market turbulence, raising asset devaluation and financial market unrest. Additionally, high inflation often leads to high-interest rates, especially where inflation is the crucial driver for monetary policy, which can reduce demand for credit and loans. This can decrease profits for financial institutions, as they earn money through interest payments on loans. Inflation can also cause liquidity challenges for financial institutions, as depositors may withdraw their funds to invest in other assets or currencies, draining financial institutions' liquidity [Yilmazkuday, 2022].

The challenges of elevated interest rates and liquidity constraints are particularly pronounced in inflation-targeted regimes. Recent financial turmoil, such as the downfall of Silicon Valley Bank, Signature Bank, and First Republic Bank in the US and decreased confidence in Credit Suisse, spotlight the risks when constricted monetary and fiscal conditions clash with inherent vulnerabilities. This turbulence, stemming from rapid interest rate hikes globally, was magnified by the pervasive influence of modern technology and the instantaneous flow of information in today's digital age. Incidents that began as localized trembles in the US banking arena rapidly rippled across the global financial sector, inducing a mass liquidation of high-risk assets and recalibrating expectations of monetary policies [imf, 2023].

At their core, financial institutions are naturally leveraged entities that use third-party capital to make a profit, and inflation negatively impacts them. Hence, accurate inflation forecasting plays a crucial role in helping risk managers make decisions and is vital in the effectiveness of inflation-targeting policies in countries like Brazil. Therefore, we emphasize the topic's importance for a research project and its alignment with the constant search for better practices adopted in the financial industry.

1.5 Objectives

Since this is an empirical study, our primary goal is to identify the most effective nonlinear machine learning model in predicting the monthly IPCA using a previous feature selection (variable selection) method based on shrinkage methods to choose the most important predictors. In addition, our study aims to analyze the prediction results using an model-agnostic explainable artificial intelligence (XAI) method called Shapely Value, which may provide insight into the ML model predictions, often deemed 'black box'.

1.6 Work Structure

The rest of this dissertation is organized as follows:

Chapter 2 : *Theoretical Basis.* A systematic literature review is conducted to better direct research efforts and ensure that relevant works have been considered.

Chapter 3 : *Machine Learning Models.* This section showcases the models used and explains how they work.

Chapter 4 : *Methodology.* Presents and discusses briefly the methodology proposed in this study.

Chapter 5 : *Empirical Analysis.* Provides an empirical analysis. The methods described in Chapter 3 are compared in terms of the prediction performance

Chapter 6 : *Machine Learning Models Interpretability.* Presents and applies the model-agnostic technique, which interprets machine learning models results.

Chapter 7 : *Conclusions.* Presents conclusions based on the results from previous chapters and outlines existing challenges.

Chapter 2

Theoretical Basis

A comprehensive literature review is an indispensable aspect of academic research, encompassing a thorough evaluation and synthesis of existing scholarly materials on a specific topic. Given the abundance of scientific information available, various fields of knowledge continue to evolve, fostering research and development endeavors to address diverse challenges. Consequently, it is imperative to ensure the pertinence of the research subject and thoroughly examine significant works right from the outset of a research project to avoid investing time and resources in studies that do not contribute significantly to the advancement of scientific knowledge. Additionally, the papers need to be scrutinized for their capacity to shed light on and effectively interpret the issues raised in the proposed study.

2.1 Method description

To determine the optimal literature references for this study, we adhere to a set of principles outlined by [Mariano & Santos, 2017]. These principles advocate for considering journal impact criteria, author citations and their associations, and the frequency of relevant keywords during the selection process. According to these authors, we need to identify the most pertinent references for the research, and that is done through a three-step approach:

1. Preliminary Research Steps: consists of defining keywords related to the topic of research, the period of analysis, the databases used, and the areas of knowledge that will be considered.
2. Presentation and interrelationship of data: consists of relating numerous sources of information, at the researcher's discretion, such as the evolution of the theme from

year to year, the authors most cited, and journals that publish the most, among others.

3. Detailing, integrative model, and evidence validation: in this step are identified the main authors, approaches, and lines of research related to the theme, using co-citation and coupling techniques.

Our intention in this project is not to implement the complete methodology proposed by [Mariano & Santos, 2017]. Rather, we plan to utilize certain aspects of the methodology to choose the most crucial studies about our objectives. These selected studies will then serve as our work's primary references and will be addressed in the literature review topic.

2.1.1 Preliminary Research Steps

Based on its widely recognized operational excellence and its suite of tools well-suited for conducting thorough literature analysis. we decided to use the data offered by the Web of Science (WoS) platform as a base for our literature review. WoS provides the advantage of data consolidation and extraction, and its extensive database includes studies dating back to 1900, ensuring comprehensive coverage of the research topic. Data collection for this study was conducted on May 14, 2023. Despite the limitation of the platform only including publications in English, this aspect has not been considered a hindrance for this research due to the broad scope of inflation forecasting, which encompasses diverse fields such as economics, mathematics, business, social science, and computer science. Based on our goals, we decided to select the following words to search:

- Inflation
- Forecasting
- Machine Learning
- Variable Selection (Feature Selection)
- Model interpretability

A clarification is necessary here. In our work, we use the term "Feature Selection" to describe the process of selecting features. However, when searching on WoS, we found more results for the term "Variable Selection" in association with "inflation" and "forecasting" compared to using "Feature Selection". Therefore, we adopted "Variable Selection" for our WoS searches. However, for the remainder of this work, we will use these terms

Table 1: Individual search of selected terms in the Web of Science database.

Termos	Publicações
"Machine Learning"	366.578
Forecasting	264.145
Inflation	76.205
"Variable Selection "	20.637
"Model interpretability"	938

Source: author

interchangeably. Table 1 displays the number of publication hits for each term in the Web of Science database without applying any filters.

Several combinations of these terms were implemented to identify the primary references that align with the objectives of this research. The keyword combinations were generated according to the guidelines presented in 2. At this stage, we decided to restrict the analysis period (from 2017 to 2023) and Research Areas (Business Economics, Computer Science, Mathematics, Mathematical Methods in Social Sciences); the results are shown in table 2.

Table 2: The keywords combinations searched in the Web of Science database.

Search	Keywords combinations	Publications
1	"Machine Learning" and "Forecasting"	14.380
2	Inflation and Forecasting	3.864
3	"Machine Learning" and Inflation	299
4	Inflation and "Machine Learning" and Forecasting	45
5	Inflation and "Variable Selection" and Forecasting and "Machine Learning"	8
6	Inflation and "Machine Learning" and Forecasting and "Model Interpretability"	2

Source: author according to the TEMAC methodology.

Table 2 shows the search resulted in many publications among the permutations performed. However, in some combinations, a low number of publications was found. So, we chose combinations four, five, and six to select our primary literature. Consolidating these results into a more concise list was necessary. Then, studies unrelated to this work's objectives were eliminated. After that, we excluded papers without at least one citation from Journals. Therefore, it was deleted documents from conferences and other sources.

Finally, the works selected for our literature review are cited in the following subsections according to our goals.

2.2 Literature Review

2.2.1 Machine Learning and Inflation Forecast

Machine learning models have grown beyond computer science and are becoming increasingly popular in many areas. The surge in data science is due to the abundance of available data thanks to advances in information technology, and this has resulted in the emergence of a new research field. Since the empirical revolution in economics during the 1970s, there has been a strong emphasis on empirical work that uses data, making econometrics a solid foundation for economists. It has given these professionals the necessary skills to explore other statistical techniques, including machine learning methods. But unfortunately, economic experts have not yet fully adopted these methodologies, unlike other fields of knowledge [Athey & Imbens, 2019].

Much of that has been due to some economists thinking that machine learning techniques are an opposing approach to econometrics based on economic theory. But machine learning can, in certain situations, outperform traditional econometrics models, as demonstrated by [Iskhakov et al., 2020]. The authors point out that, when it comes to predictions, machine learning focuses on how computers make such predictions, presenting a more practical orientation, while structural econometrics focuses on understanding how humans make these predictions, characterized as a more academic approach.

Trying to help economists, [Athey & Imbens, 2019] synthesized machine learning methods and how these methods can benefit econometrics and economics research. Machine learning presents distinct objectives, approaches, and environments in contrast to conventional statistics and econometrics. They draw attention to specific ML methods vital for empirical research, such as supervised, an approach that aims to estimate the conditional mean of a variable based on a broad set of covariates, and unsupervised learning, matrix completion, and newer ways that perform better than off-the-shelf ML or traditional econometric methods in certain areas. Areas of focus include causal inference, optimal policy estimation, and estimating counterfactual effects on consumer choices due to price changes.

Regarding models applied in time series, [Masini et al., 2021] presented various ML techniques. They use both linear and non-linear models and hybrid methodologies that prove advantageous in applications. They present mathematical formalizations of various models and emphasizes the remarkable results of the penalized regression model and the hybrid approach. [Nosratabadi et al., 2020] explore various machine learning techniques, particularly those with a hybrid nature, and demonstrate their effectiveness in various economic applications like stock price prediction, consumer behavior for marketing purposes and cryptocurrency market seeks to forecast the prices of digital currencies. As the authors pointed out, hybrid models combine two ML algorithms for prediction or integrate an ML algorithm with an optimization method to maximize the prediction function. The authors conduct a systematic review of machine learning methodologies and present the applications found in the literature for economics.

Regarding inflation forecasts using ML techniques, [Behrens et al., 2018] examines the optimality of macroeconomic forecasts. They used random forests, a powerful nonparametric modeling instrument, to test the prediction that variables available to a forecaster should have no predictive value for a binary 0/1-indicator that captures the sign of the forecast error. According to them, Random forests are a powerful modeling device for studying macroeconomic forecasts and perform better than conventional linear probability or logit/probit-models over nonlinear problems. They account for nonlinear links between forecast error and variables and can handle small forecast numbers relative to predictor variables. They analyzed inflation forecasts from four German research institutes and found that longer-term forecasts were optimal and efficient. However, shorter-term forecasts varied between institutes, and overall, neither short nor long-term forecasts were optimal when the data was pooled.

The study by [Medeiros et al., 2021] is based on inflation data from the United States economy. It uses machine learning to perform forecasts, which proved to be much more accurate than conventional methodologies, especially when many covariates are involved. The authors highlight the superior performance of the random forest model, which corroborates the consensus of the literature that studies forecasts of macroeconomic series. This class of models allows exploring non-linear relationships between macroeconomic variables and inflation. [Alfiyatin et al., 2019] propose using an extreme learning machine (ELM) to forecast inflation, with particle swarm optimization (PSO) to determine good initial weights for the ELM. Our approach achieves better results than the original ELM.

Our study is dedicated to applying ML approaches to forecasting inflation in Brazil. Therefore, we need to see what other authors from emerging and developed economies are creating and how they deal with inflation forecasts and ML techniques. A common approach among studies with a similar goal is to compare the predictive accuracy of ML techniques to traditional econometric methods. The literature has generally shown examples where ML methods surpass traditional approaches. For example, [Riofrío et al., 2020] conducted a study on predicting Ecuador's CPI using state-of-the-art models. They tested models including Neural Networks, Support Vector Regression, SARIMA, and Exponential Smoothing. They conclude that the Support Vector Regression (SVR) model using a polynomial kernel is the best element to predict the Ecuadorian CPI one year ahead. [Syed & Lee, 2021] compares forecasting models for Pakistan's CPI inflation, GDP growth, and overnight repurchase rates. They use 161 predictors and evaluate the performance of benchmark models and machine learning approaches. Their results showed that ML approaches outperform the benchmark and commonly used Dynamic factor models (DFM). [Acosta, 2018] proposed a new methodology for developing core inflation measures, which utilizes the k-means clustering machine learning algorithm and aimed to obtain a clear signal and accurate predictions of the inflationary process through selecting items with low volatility and assigning them to clusters. They showed that the core inflation produced using this methodology captures the inflation signal more effectively and performs better than the short-term inflation forecasts obtained through the trimmed means method and the core inflation that excludes food and energy.

Moving forward, let's focus on predicting the consumer price index in Brazil. [Garcia et al., 2017] use machine learning models to estimate real-time Inflation. According to them, this approach is necessary due to the high short-term volatility of this variable in emerging countries. The authors use the predictions made by experts as a reference and comparison for the prediction models, which include the factor model, LASSO/adaLASSO, random forests, and complete subset regression. Model performance varies significantly with the desired forecast horizon. The LASSO model, for example, performed similarly to the FOCUS report for forecasting periods of $t+1$, while adaLASSO proved to be more efficient for a horizon of $t+2$. The best result was obtained when building a forecast that combines the average of the models' results, surpassing all alternatives.

[Araujo & Gaglianone, 2023] have successfully predicted Brazilian headline IPCA using ML methods. They claim that based on empirical results, machine learning methods can outperform traditional econometric models in multiple cases. Additionally, the results suggest the presence of non-linearities in inflation dynamics that are significant for predicting inflation. Leading machine learning predictions often utilize a blend of forecasting techniques, incorporating ensemble methods like random forest and XGBoost and breakeven inflation rates and survey expectations.

Table 3 summarizes which models were used by each author dealing with the intersection between inflation forecasting and ML methods.

Table 3: Summary of ML Models applied to Inflation Forecasting

Authors	ML Models	Target Series	Countries
[Aras, 2022]	ExtraTrees, Adaboost, Gradient Boosting and XGBoost.	Inflation	Turkey
[Joseph et al., 2021]	Ridge, LASSO, ElNet, NN, SVM, Random Forest, PLS, PCA	Inflation	UK
[Coulombe et al., 2022]	Kernel Ridge Regression, Random Forest, Ridge, Lasso, ElNet	Industrial Production, Unemployment, Inflation, Spread Treasuries, Housing Start	USA
[Buckmann et al., 2021]	Random Forest, NN, Linear Regression, Ridge, SVR, AR	Inflation, Unemployment	USA
[Özgür & Akkoç, 2021]	VAR, ARIMA, Ridge, Lasso, adaLasso, ElNet	Inflation	Turkey
[Acosta, 2018]	k-means clustering	Inflation Core	Mexico
[Syed & Lee, 2021]	DFM, Ridge regression, LASSO ElNet, Bagging Methods	GDP, Inflation	Pakistan
[Medeiros et al., 2021]	LASSO, adaLASSO, EINet, adaEINet, Ridge, BVAR, Random Forest	Inflation	USA
[Garcia et al., 2017]	Factor model, LASSO, adaLASSO, Random Forest, full subset regression, CSR	Inflation	Brazil
[Riofrío et al., 2020]	Neural Networks, Support Vector Regression, SARIMA, and Exponential Smoothing	Inflation	Ecuador
[Alfiyatin et al., 2019]	Extreme Learning Machine (ELM)	Inflation	Indonesia
[Behrens et al., 2018]	Random Forest	inflation	Germany
[Araujo & Gaglianone, 2023]	RNN, Random Forest, XGBoost, LASSO, Ridge, ElNet, Hybrid models, Factor Model, Phillips Curve, VAR, ARMA	Inflation	Brazil

Source: Author.

2.2.2 Machine Learning and Feature Selection

As per a study conducted by [Fonti & Belitser, 2017], feature selection, also called variable selection, involves selecting a smaller set of explanatory variables to describe the problem at hand. This helps algorithms to work faster and makes it easier to handle high-dimensional data. The reasons why feature selection is used are:

1. Simplify the model by eliminating unnecessary variables that do not contribute any useful information.

2. By decreasing the problem's size, algorithms can operate more quickly, allowing for the processing of high-dimensional data.

3. Mitigates overfitting: Removing non-essential features simplifies the model and curtails the tendency to fit excessively to the noise in the training data.

4. Improves Accuracy: Sometimes, less relevant or noisy features can cause models to overfit to the training data.

In high-dimensional datasets, like we are working in our study, variable selection becomes particularly crucial. In such cases, the number of features is often significantly greater than the number of observations. This makes it challenging to determine the relevance of each variable and distinguish between the relevant and irrelevant ones.

According to [Stańczyk & Jain, 2015], common Techniques for Feature Selection are:

1. **Filter Methods**
   (a) Correlation Coefficient: Correlates each feature with the target variable and removes low correlation features.
   (b) Chi-Squared Test: Used for categorical features, it checks the dependence of one feature on the other.
   (c) Information Gain: Measures the reduction in entropy brought about by partitioning on a given feature.
   (d) Variance Threshold: Features with variance below a certain threshold can be removed, as they might be considered constant and therefore not informative.
2. **Wrapper Methods**:
   (a) Recursive Feature Elimination (RFE): Fits the model multiple times and in each iteration, removes the weakest features.
   (b) Sequential Feature Selector: Sequentially adds or removes features to find the best subset.
   (c) Backward Elimination & Forward Selection: In backward elimination, you start with all features and remove one feature in each iteration, while in forward selection, you start with no features and add one in each iteration.
3. **Embedded Methods**:
   (a) Shrinkage Method: the model uses L1 regularization to select features by shrinking some coefficients to zero.
   (b) Tree-based Models: Decision Trees, Random Forests, Gradient Boosted Trees, etc., provide importance scores for features. Features with low importance can be dropped.
   (c) Regularization Methods: Models like Ridge and Elastic Net regularization can also help select feature by adding penalty terms.

4. **Hybrid Methods**: These are combinations of the above methods to leverage the strengths of different methods. For example, a filter method can be used to reduce dimensionality before applying RFE.

5. **Dimensionality Reduction as Feature Selection**:

   (a) Principal Component Analysis (PCA): Creates new orthogonal features called principal components, which are linear combinations of the original features. However, these new features are interpreted differently than the original ones.

   (b) Linear Discriminant Analysis (LDA): Seeks to maximize class separability.

   (c) Regularization Methods: Models like Ridge and Elastic Net regularization can also help select feature by adding penalty terms.

[Stańczyk & Jain, 2015] highlight that feature selection is data-dependent. Just because a feature is deemed unimportant for one data-set doesn't mean it will be insignificant for another. Domain knowledge can be invaluable. Sometimes, domain experts can point out important features or irrelevant ones before any algorithmic feature selection begins.

In conclusion, feature selection is a crucial step in the machine learning process. It simplifies models, speeds up training, and improves performance. We can create more efficient, interpretable, and accurate models by selecting the most relevant and essential features.

Looking specifically at the intersection between inflation forecasting and feature selection (variable selection), [Kim & Swanson, 2018] analyzed the benefits of using principal component analysis (PCA), independent component analysis (ICA), and sparse principal component analysis (SPCA) for forecasting macroeconomic variables such as unemployment, inflation, and GDP. They also evaluate machine learning, variable selection, and shrinkage methods, including bagging, boosting, ridge regression, least angle regression, the elastic net, and the non-negative garotte. According to the authors, factor-based dimension reduction is helpful for macroeconomic forecasting with "big data."

[Özgür & Akkoç, 2021] using machine learning algorithms to predict inflation rates in Turkey, comparing them to traditional econometric models. Shrinkage methods were more effective in feature selection, with the lasso and elastic net algorithms outperforming other methods. The authors found these algorithms can identify essential variables for predicting inflation, specifically energy production, construction-sector measures, exchange rates, and money market indicators.

2.2.3 Machine Learning Interpretability

Forecasting inflation accurately is difficult, particularly in a data-rich environment with highly correlated predictors. While various feature selection methods attempt to address this, they often fail to explain how computers make their predictions. The interpretability of machine learning models is crucial in many fields, including inflation forecasting. Interpreting these 'black box' models holds significant importance since it can assist economic agents in navigating inflation stress scenarios. The Shapley Value method is employed in this work as a model-agnostic tool for interpreting these models. Therefore, reviewing literature that works with inflation forecasting and machine learning interpretability is valuable.

To address the challenge of machine learning models being limited in terms of result interpretability, particularly in macroeconomic theory, [Coulombe et al., 2022] aim to establish a link between machine learning methods and conventional macroeconomic approaches for various macroeconomic variables, including the US CPI. Although the authors didn't work with Shapley Value they identified and tested four features driving ML gains in data-rich and data-poor environments: non-linearity, which, despite the prevalence of linearized models, offers advantages by allowing forecasts with less error when the data-generating process is also non-linear; regularization, which makes it possible to reduce the complexity of the estimation when there are many variables involved; cross-validation, which acts as a selection criterion similar to the Akaike and Schwarz-Bayesian information criteria, frequently used in macroeconometrics; and the exploration of alternative loss functions since the quadratic error function is the most commonly used. Their findings show that nonlinearity is the game changer, the standard factor model remains the best regularization, K-fold cross-validation is the best practice, and L2 is preferred to e-insensitive in-sample loss. Still, according to the authors, ML captures critical nonlinearities in uncertain and financially stressful situations, making it useful for macroeconomic forecasting.

[Joseph et al., 2021] use ML techniques to predict inflation in the UK up to one year ahead. They improved forecasting accuracy beyond macroeconomic predictors using monthly CPI item series data. The authors also offer a model-agnostic approach for interpreting high-dimensional models using statistical testing and Shapley values. This way, they could identify CPI divisions that consistently drive forecasts and assess model differences beyond accuracy.

[Buckmann et al., 2021] used nine sets of US economic data to predict and compare the results of machine learning models with traditional time series forecasting methodologies. In addition, they applied interpretability techniques to understand better the results obtained. Random forest models showed superior results. Using the Shapley regression technique, they concluded that variables such as industrial production and the S&P 500 played a relevant role in predicting the unemployment rate in the United States.

A study by [Aras, 2022] recommends using ML models with feature selection techniques for better accuracy and explainable predictions. The writer employed Shapley values to obtain succinct explanations of inflation forecasts to achieve their objective. Their research involved conducting numerous experiments in Turkey, a country with high levels of volatility and uncertainty. The outcomes revealed that tree-based ensemble models can offer improved accuracy and comprehensible predictions.

Chapter 3

Models

This section introduces the machine learning models employed in this study. We aim not to provide an exhaustive and comprehensive presentation of each model. Instead, we focus on the key concepts and equations and provide relevant references for those interested in a more detailed exploration.

We begin our discourse by exploring the regularized linear models, which will be used to perform the feature selection (variable selection) in this work. Subsequently, we introduce nonlinear ML models.

3.1 Regularized linear models

Linear regression models are often estimated using Ordinary Least Squares (OLS) model estimators. This method involves finding the model with the smallest Euclidean distance between the observed and adjusted values. These estimators have optimal properties, including the Gauss-Markov theorem, which states that the least squares estimator has the most minor variance among all unbiased linear estimators. However, in some situations, OLS estimators may not be a good model estimation option.

For example, in scenarios where the number of predictors is greater than the number of observations (i.e., $p > n$), the solution found by OLS models can be problematic as it is not unique. OLS is highly susceptible to overfitting even when n approaches but is still smaller than p. This means there can be infinite solutions for the objective function, making OLS estimators suboptimal for estimating models. This scenario is frequently encountered with macroeconomic series due to their relatively brief duration and too many alternatives to measure the same variable.

Besides, [Tibshirani & Wainwright, 2016] highlights two other key motivations for seeking alternatives to the least squares estimate. Firstly, prediction accuracy: while least squares estimates are typically low in bias, they suffer from high variance. Secondly, the goal of interpretability: it becomes desirable to pinpoint a more concise subset with the most pronounced impact in situations with numerous predictors.

To overcome these problems, some authors proposed penalized regression methods, also called Shrinkage models. In the context of equation (3.1) and the general forecasting framework, the Shrinkage methods assume that $f\left(\boldsymbol{x}_t\right)=\boldsymbol{x}_t^{\top} \boldsymbol{\beta}$, allowing us to express the general formulation of the forecasting models as:

$$y_{t+h}=\boldsymbol{x}_t^{\top} \boldsymbol{\beta}+\varepsilon_{t+h}, \quad t=1, \cdots, T \tag{3.1}$$

The group of penalized linear models adds a penalty component to the OLS objective function, aiming to regulate the complexity of the model.

$$\sum_{t=1}^{T-h}\left(y_{t+h}-\boldsymbol{x}_t^{\top} \boldsymbol{\beta}\right)^2+\lambda p(\boldsymbol{\beta}) \tag{3.2}$$

Here, $\lambda \geq 0$ represents a penalty (or tuning) hyperparameter, and $p(\cdot)$ denotes the penalty function. The solution is obtained by minimizing:

$$\widehat{\boldsymbol{\beta}}(\lambda) \in \underset{\boldsymbol{\beta} \in \mathbb{R}^N}{\operatorname{argmin}}\left[\|\boldsymbol{y}-X \boldsymbol{\beta}\|_2^2+\lambda \mathrm{p}(\boldsymbol{\beta})\right] \tag{3.3}$$

The penalty function should yield values on the positive real line, penalizing $\boldsymbol{\beta}$ for deviating from zero. Thus, the family of penalized linear models shrinks the coefficients towards zero compared to their OLS estimates. It is important to note that $\lambda \geq 0$ determines the overall strength of the penalty, where $\lambda = 0$ corresponds to the typical OLS estimator.

To estimate models using penalized regression models, we must first apply some Feature scaling technique. According to [Geron, 2019] Feature scaling is a method used to normalize the range of independent variables or features of data. Data processing is also known as data normalization and is generally performed during the data preprocessing step. We must do this so that the estimation is not affected by the unit of measurement or their order of magnitude. One covariate, for example, can be in meters, another in kilometers and a third can be dimensionless with a very different order of magnitude from the others. In a nutshell, scaling the covariates is a recommended practice when dealing with models in this category.

The study employs the ElasticNet technique to select variables. However, before delving deeper into this method, it is crucial to first understand LASSO regression. This is the most common option used to perform feature selection through shrinkage. LASSO regression has a specific condition that must be met before it can be used. Due to this limitation, we opted for ElasticNet as it has a more suitable approach for dealing with datasets containing many correlated variables.

3.1.1 LASSO Model

As we said previously, when the number of predictors p surpasses the series' length n observations ($p > n$), the solution found by OLS models can be problematic as it is not unique. Moreover, the lack of sparsity [1] can result in models with numerous parameters that are difficult to interpret, and there is a problem related to low bias but high variance. Penalized methods were created to overcome these problems, and among many alternatives using this approach, there is the Least Absolute Shrinkage and Selection Operator (LASSO) regression created to address these issues; thus, it is a useful alternative, and it is important to understand how it works.

LASSO was introduced by [Tibshirani, 1996], and its variations are distinguished by its ability to perform regularization and variable selection simultaneously. The penalty term takes the following form:

$$\lambda p(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{n} |\beta_i| = \lambda \|\boldsymbol{\beta}\|_1 \tag{3.4}$$

Similar to Ridge Regression[2], the LASSO also drives coefficients toward zero. However, the penalty introduced by the l_1 [3] norm possesses a distinct attribute: it can precisely set specific coefficients to zero when the value of λ reaches a certain magnitude. This results in what is commonly referred to as a sparse solution, wherein only a subset of the entries in $\hat{\boldsymbol{\beta}}^{\text{Lasso}}$ exhibits non-zero values when λ becomes sufficiently sizable.

In a nutshell, the regularization parameter, λ, is pivotal in LASSO regression. It controls the trade-off between fitting the data well and keeping the model simple through sparsity.

- $\lambda = 0$: The LASSO model becomes equivalent to an ordinary least squares model, with no feature selection.
- As λ increases: More coefficients are set to zero, enhancing sparsity but potentially leading to underfitting.

[1] Sparsity comes from the Latin sparsus. One definition is the property of being scanty or scattered, lacking denseness. Applied to ML field, according to [Tibshirani, 2014], sparsity refers to a property where only a small subset of features or coefficients in the model contribute significantly to the predictions. At the same time, the majority remain zero or negligible.

[2] Hoerl and Kennard (1970) pioneered Ridge Regression to deal with highly correlated predictors, highlighting an essential feature where coefficients are shrunken towards zero, yet not precisely to zero.

[3] l_1 regularization penalizes the sum of absolute values of the weights, whereas l_2 regularization penalizes the sum of squares of the weights. The l_1 regularization solution is sparse like the lasso models. Like the Ridge models, the l_2 regularization solution is non-sparse.

- Optimal λ: Found using cross-validation, represents the best balance between model complexity and fitting the data.

This research will use Elastic Net for feature selection (variable selection), which uses LASSO as a base. Therefore, it is crucial to discuss the properties of LASSO and its conditions.

3.1.2 Mechanism of LASSO Feature Selection

Model selection is essential in statistical modeling, especially with high-dimensional data. Generically, model selection refers to deciding which of a set of models is best (each encodes a different set of assumptions). To address this problem, LASSO can usefully produce some coefficients that are exactly zero, effectively selecting a simpler model that excludes some features altogether. This contrasts with Ridge Regression, where all coefficients are typically non-zero but smaller in magnitude. The key mechanism behind feature selection in LASSO regression lies in the l_1 penalty term, $\lambda \sum_{i=1}^{n} |\beta_i|$. This penalty term has two crucial effects:

1. Regularization Effect: By constraining the sum of the absolute values of the coefficients, LASSO regression controls the model complexity, thereby preventing overfitting. This is particularly vital in scenarios with many predictors or when multicollinearity is present.
2. Sparsity Effect: The absolute value in the penalty term makes the cost of including a variable in the model nonlinear in its coefficient. This nonlinearity has a unique property: as the penalty increases, coefficients of less important variables are driven precisely to zero, effectively removing them from the model. Unlike Ridge regression, another regularization technique only shrinks coefficients close to zero but never precisely to zero.

Exploring the distinctions between the penalty mechanisms of LASSO and Ridge reveals why LASSO possesses the feature selection ability while Ridge does not. Drawing on the insights from [Li, 2024] and [James et al., 2023], we delve into the optimization problems of both LASSO and Ridge to highlight their differences. The optimization for LASSO can be represented by:

$$\Sigma_{i=1}^{n} \left(y_i - \beta_0 - \Sigma_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \Sigma_{j=1}^{p} |\beta_j| = \tag{3.5}$$

$$\min_{\beta} \left\{ \Sigma_{i=1}^{n} \left(y_i - \beta_0 - \Sigma_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\}, \Sigma_{j:}^{p} \tag{3.6}$$

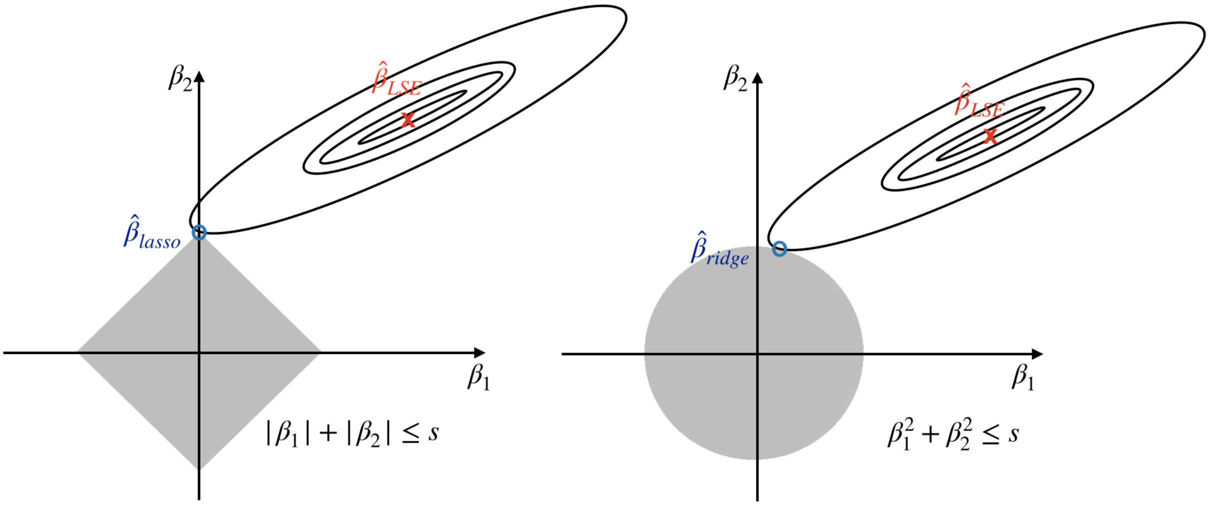
For any given λ, there's a corresponding s that ensures the solutions optimizing equation 3.5 also satisfy equation 3.6. Ridge regression's optimization problem similarly aligns across two representations:

$$\Sigma_{i=1}^{n}\left(y_i-\beta_0-\Sigma_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\Sigma_{j=1}^{p}\beta_j^2= \tag{3.7}$$

$$\min_{\beta}\left\{\Sigma_{i=1}^{n}\left(y_i-\beta_0-\Sigma_{j=1}^{p}\beta_j x_{ij}\right)^2\right\},\Sigma_{j:}^{p} \tag{3.8}$$

[Li, 2024] elucidates that for $p=2$, LASSO's estimates achieve the lowest RSS within all points fulfilling $|\beta_1|+|\beta_2|\leq s$. Conversely, Ridge's estimates are optimal within a boundary where $\beta_1^2+\beta_2^2\leq s$. As s expands, so do the diamond and circle regions, becoming less restrictive. When s is sufficiently large, these regions encompass the least squares estimate $\left(\hat{\beta}_{\text{LSE}}\right)$, allowing equations 3.6 and 3.8 to converge to the least squares estimate. Conversely, a smaller s limits the size of the grey region in figure 3.1, thus constraining β's magnitude. This alternate formulation provides a clear explanation for the feature selection attribute inherent in LASSO for feature selection.

Figure 3.1: Contours of the RSS and constrain functions for the lasso (left) and ridge regression (right);



Source: [Li, 2024].

Some benefits of using LASSO for feature selection are turning models interpretable and reducing overfitting. For the first case, LASSO helps identify the essential features, making the model more understandable and less complex in coefficient amount. Related to the second point, LASSO can mitigate overfitting by reducing the coefficients of less relevant features to zero, resulting in a more robust model. Besides, this technique can handle correlation among features. When features are correlated, LASSO tends to choose one of them based on its performance in the given dataset, which can simplify the model without losing much predictive power. However, there are some disadvantages when using the LASSO approach to perform a feature selection.

3.1.3 LASSO Sparsity Condition

The LASSO feature selection method is intuitively simple in both theory and practice. It offers advantages such as enhanced interpretability and reduced overfitting by eliminating irrelevant predictors. However, its efficacy can falter in highly correlated datasets. LASSO's selection of predictors may lack consistency across different runs or datasets, particularly when predictors exhibit strong correlations, demanding cautious interpretation, especially in the presence of multicollinearity. This inconsistency raises questions about its reliability as a model selection technique. Authors such as [Özgür & Akkoç, 2021] and [Sklearn, 2024] highlight this drawback not only in LASSO but also in Ridge regression. When explanatory variables are strongly correlated, the LASSO estimator may treat them indifferently, whereas Ridge regression tends to shrink them collectively. Additionally, as shown by [Wang et al., 2020] in their simulation, l_1 regularization tends to introduce many spurious variables in very sparse cases. These issues can yield unexpected outcomes with significant practical implications.

According to [García-Portugués, 2024] and [Zhao & Yu, 2006], the LASSO method consistently identifies the correct model under the condition that the predictors meet a specific requirement, termed the strong irrepresentable condition. Additionally, the regularization parameter $\lambda = \lambda_n$ must approach zero at certain speeds as $n \to \infty$.

Despite its technical nature, the strong irrepresentable condition requires that correlations among predictors be managed within certain bounds. [Zhao & Yu, 2006] outlined a few straightforward scenarios in which this condition is fulfilled:

1. When predictors are not correlated.

2. When predictor correlations are kept within a limit as delineated by the authors. Specifically, if β^{-1} contains q non-zero entries, it must hold that $\mathrm{Cor}[X_i, X_j] \leq c2^{q-1}$ for a constant $0 \leq c < 1$, for all $i, j = 1, \ldots, p$, with $i \neq j$.

3. When correlations among predictors follow a power law, as outlined by the authors. That is, $\mathrm{Cor}[X_i, X_j] = \rho^{|i-j|}$, for $|\rho| < 1$ and for all $i, j = 1, \ldots, p$.

Despite the clarity these conditions provide on LASSO's consistency in model selection under specific circumstances, they offer little insight into its reliability in more complex scenarios, particularly when the penalization parameter $\hat{\lambda}$ is determined through data-driven methods (as opposed to the deterministic sequence $\lambda_n \to 0$, as specified by [Zhao & Yu, 2006]).

To address these challenges, alternative shrinkage methods have emerged. For instance, Elastic Net strikes a balance by inducing sparsity in coefficients while also shrinking their values toward zero. This approach enables the model to reduce the weights of correlated features without precisely setting them to zero, resulting in a less sparse model than pure LASSO. By incorporating a penalty function with two tuning terms, Elastic Net encourages sparse variable selection while promoting coefficient averaging. While these adjustments mitigate some challenges, ongoing research aims to refine models for better performance in handling highly correlated datasets.

3.1.4 Elastic Net Model

So, the Elastic Net, put forth by [Zou & Hastie, 2005], introduces a penalty component that merges the penalties of Ridge and LASSO using a convex blend:

$$\lambda p(\beta) = \lambda \left[(1 - \alpha) \|\beta\|_2^2 + \alpha \|\beta\|_1 \right] \tag{3.9}$$

In this method, the hyperparameter λ controls the overall strength of the penalty, while a new hyperparameter α determines the relative importance of the l_1 and l_2 penalties. Notably, when $\alpha = 0$, the penalty reduces to the ridge regression, and when $\alpha = 1$, it becomes equivalent to the LASSO. This modified penalty term provides an improvement over the LASSO in scenarios involving highly correlated predictors. It encourages a grouping effect, where strongly correlated predictors are more likely to shrink to zero or be excluded from the model.

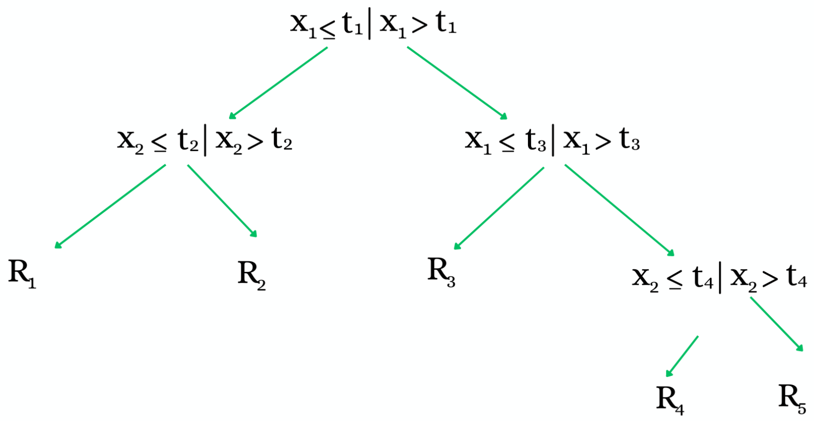
LASSO and Elastic Net Regression stand out for feature selection within complex datasets, primarily due to their capacity to induce sparsity. Elastic Net, in particular, is favored for its proficiency in handling correlated features, a common trait in our time-series dataset, leading us to use it for variable selection. This technique adeptly negotiates between model precision and complexity, a balance vital in statistical modeling and machine learning contexts. However, the final model interpretation should always consider domain knowledge and context. These methods are beneficial in high-dimensional data scenarios where understanding variable relationships is critical.

3.2 Nonlinear Models

ML models approaches are recognized for their "inductive" nature, allowing them to adapt organically to sampled data and derive decision-making functions from observed patterns without relying on predetermined functional shapes or making specific assumptions about the data's distribution. Some authors support using machine learning techniques in traditional economics approaches, highlighting their enhanced empirical predictive power and proficiency in uncovering nonlinear dynamics. In this sense, our study uses regression tree-based methods as nonlinear models. These methods involve dividing the predictor space into simple regions using tree structures. We then make predictions by taking the average or mode of the training observations within these regions. These techniques are called 'decision trees' because the rules used to partition the predictor space are represented by trees. With few predictors, tree-based methods are simple and easy to understand. However, we use multiple trees to enhance prediction accuracy, which may affect interpretability.

Figure 3.2 illustrates the binary subdivision of regions in a scenario where one wishes to analyze a variable Y concerning two others, X_1 and X_2. The subdivision of regions continues until a stopping condition is triggered. Initially, the tree covers the entire available dataset. Then, the division of areas is based on the condition $X_1 = t_1$. If $X_1 \leq t_1$, the left branch is followed; otherwise, the right unit is taken. This process is repeated sequentially, resulting in five distinct regions: $R_1, R_2, \ldots, R_5$.

Figure 3.2: Regression tree.



Source: author.

The equation that describes the regression for estimating Y, using a constant c_m representing the mean or median of the observations in region R_m, is expressed as follows:

$$\widehat{f}(X) = \sum_{m=1}^{5} c_m l \{(X_1, X_2) \in R_m\} \tag{3.10}$$

To sum up, we aim to recursively predict an unknown nonlinear function using regression trees to divide the covariate space. Starting at the root node, a subset of the variable set, we create two child nodes for each non-terminal node. We identify the best split using an algorithm and apply a Boolean condition to the variable value. If a node cannot be split, we create a terminal node or leaf, and the value on the terminal node determines the output variable.

The tree size is a tuning parameter to determine the model's complexity. Large trees can overfit the data, while small trees may not capture the data's structure. Therefore, it is crucial to select the optimal size adaptively based on the data.

3.2.1 Random Forest

The Random Forest approach, which first introduced [Breiman, 2001], is utilized for classification and regression purposes. For regression, the algorithm creates a set of regression trees by selecting a bootstrapped sub-sample of the initial data for each tree. When growing the trees, the algorithm only considers a random subset of variables at each node to determine the optimal split. The forecast for each tree is made at the terminal nodes, and the final prediction is calculated by averaging the results of all the trees. In simpler terms:

$$\hat{y}_{t+h} = \frac{1}{B} \sum_{b=1}^{B} f_b\left(\mathbf{x}_t\right) \tag{3.11}$$

We aim to forecast the target feature y_{t+h} using B regression trees. The output of each tree, $f_b\left(\mathbf{x}_t\right)$, is based on a bootstrapped sub-sample of the training data and the randomization of the variables considered to determine the best split at each node.

Bagging involves taking an average of models created from multiple bootstrap samples to enhance the performance of an estimator. This method effectively improves the accuracy of certain estimators, particularly nonlinear ones like trees and neural networks [Friedman & Hall, 2007]. Random forests take this further by introducing randomness to the tree-growing process, reducing the correlation between individual trees, and increasing effectiveness. Considerable evidence supports the effectiveness of bagging and random forests in improving estimators [James et al., 2013].

Random forests are prevalent when we talk about inflation forecast, as table 3 shows, and can perform remarkably well in representing complex relations in the data, with minimal tuning required when compared to methods like deep neural networks [Athey & Imbens, 2019].

3.2.2 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems proposed by [Friedman, 2001] that involves constructing additive models by fitting a base learner to the current pseudo-residuals at each iteration. Pseudo-residuals refer to the gradient of the loss function that the model tries to minimize, evaluated at each training data point and current step. In simpler terms, given a dataset $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where x_i represents the features and y_i is the corresponding target value, regression aims to find a function $f(x)$ that maps x to y.

1. **Weak Learners**: In gradient boosting, an ensemble is progressively enhanced by incorporating predictors that address the errors of their predecessors. These predictors are usually simple, such as decision stumps (single-split trees), and are termed "weak learners" due to their marginal enhancement beyond mere chance.

2. **Gradient Descent**: Gradient boosting diverges from AdaBoost's strategy of adjusting instance weights with each iteration, choosing instead to train new predictors on the residual errors left by the preceding one. This approach resembles a numerical optimization challenge, employing gradient descent to reduce a specified loss function.

According to [Friedman, 2001], an algorithm for Gradient Boosting Regression works as follow:

1. Start the model with a fixed value, $f_0(x)$:

$$f_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3.12}$$

Where L is the loss function. For the sake of simplicity, we'll assume the mean squared error (MSE) for regression:

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2 \tag{3.13}$$

2. For each stage $m = 1$ to M, do the following:

a. Compute the pseudo-residuals:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}} \tag{3.14}$$

For MSE, this reduces to:

$$r_{im} = y_i - f_{m-1}(x_i) \tag{3.15}$$

b. Fit a weak learner $h_m(x)$ to the pseudo-residuals.

c. Compute the multiplier γ_m that minimizes the loss:

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + \gamma h_m(x_i)) \tag{3.16}$$

d. Update the model:

$$f_m(x) = f_{m-1}(x) + \gamma_m h_m(x) \tag{3.17}$$

3. The final model is:

$$f(x) = f_M(x) = f_0(x) + \sum_{m=1}^{M} \gamma_m h_m(x) \tag{3.18}$$

When predicting inflation using gradient boosting, as we intend to do in this work, the features x might include economic indicators such as GDP growth rate, unemployment rate, previous inflation rates, monetary policy rates, and others. The target value y would be the inflation rate. Given the non-linear and often complex relationships between economic indicators and inflation, gradient boosting is an apt choice as it can capture intricate patterns in the data, provided the model is tuned properly. In conclusion, gradient boosting is a powerful ensemble method that can be employed for predicting inflation rates using regression.

3.2.3 Extreme Gradient Boosting - XGBoost

Using gradient boosting techniques, [Chen & Guestrin, 2016] introduced a scalable machine learning system known as XGBoost. This system utilizes gradient boosting for regression trees. This model's structure closely mirrors that of traditional gradient boosting, focusing on loss minimization while incorporating a complexity-based regularization term (Ω). The simplified version of the objective function for the t-th iteration is given as:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t \left(\mathrm{x}_i\right) + \frac{1}{2} h_i f_t^2 \left(\mathrm{x}_i\right) \right] + \Omega \left(f_t\right), \tag{3.19}$$

where
$g_i = \partial_{\widehat{y}^{(t-1)}} l \left(y_i, \hat{y}^{(t-1)}\right)$
and
$h_i = \partial^2_{\widehat{y}^{(t-1)}} l \left(y_i, \widehat{y}^{(t-1)}\right)$
are the first and second-order gradients of the cost function.
Given that
$\Omega(f) = \gamma T + \frac{1}{2}\lambda|w|^2$,

we expand Equation (35) by defining the instance set
$l_j = (i|q(\mathbf{x}_i) = j)$ for leaf j:

$$\begin{aligned}\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n} \left[g_i f_t\left(\mathrm{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathrm{x}_i\right) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \\ &= \sum_{j=1}^{T} \left[\left(\sum_{i \in l_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in l_j} h_i + \lambda \right) w_j^2 \right] + \gamma T,\end{aligned} \tag{3.20}$$

for a fixed structure $q(\mathbf{x})$, the weight w_j^* for leaf j is given by:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \tag{3.21}$$

which implies an optimized value for the problem that can be used as a performance measure for the quality of the tree q, defined as:

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{3.22}$$

Due to the complexity of enumerating all possible structures q, an algorithm is applied that starts with a leaf and adds branches to the tree. Assuming that I_L is the left node after the split and I_R is the right node, where $I = I_L \cup I_R$, the loss reduction after the split, which is used to find potential candidates for splitting, is defined as:

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma. \tag{3.23}$$

The XGBoost model employs three distinct algorithms for identifying split points:

1. The first is the greedy algorithm, which seeks to find a point of global optimum using first and second-order gradients, which can be computationally expensive.

2. The second is an approximate algorithm that uses an initial split based on percentiles, making comparative analyses between potential split points.

3. The third is an algorithm capable of handling sparsity issues, where some of the covariates do not have values for all instances compared to other covariates.

As a result, XGBoost can apply greedy algorithms, allowing it to find approximations of local or global optimal points. It supports out-of-core processing, which is very useful for conserving machine memory, especially when dealing with very large datasets. XGBoost can handle data sparsity issues and also parallelization problems, where a block system reduces the computational costs associated with sorting database variables, particularly for very large datasets.

3.2.4 Adaptive Boosting

Adaptive Boosting, commonly abbreviated as AdaBoost, operates within the realm of machine learning algorithms, specifically targeting the enhancement of weak learners. This methodology is distinctive for its adaptability, particularly in addressing the challenges posed by instances that were incorrectly classified by preceding classifiers. AdaBoost exhibits a notable sensitivity towards noisy data and outliers, yet it demonstrates a reduced susceptibility to overfitting in comparison to various other learning algorithms. Despite the fundamental weakness of individual learners, the collective performance—provided it marginally exceeds random guessing—cumulatively fosters the development of a robust learner through iterative convergence.

AdaBoost is notably employed in the context of training boosted classifiers. A boosted classifier embodies a composite model represented by the equation:

$$F(x) = \sum_{T=1}^{T} f_t(x) \tag{2.6}$$

Here, $f_t(x)$ signifies a weak learner that accepts an object's input x and outputs a value indicative of the object's class affiliation. Specifically, in a binary classification scenario, the sign of $f_t(x)$'s output delineates the predicted class of the object, whereas the output's absolute value reflects the confidence in this classification. The T^{th} classifier assigns a positive label if the sample belongs to the positive class, and a negative label otherwise.

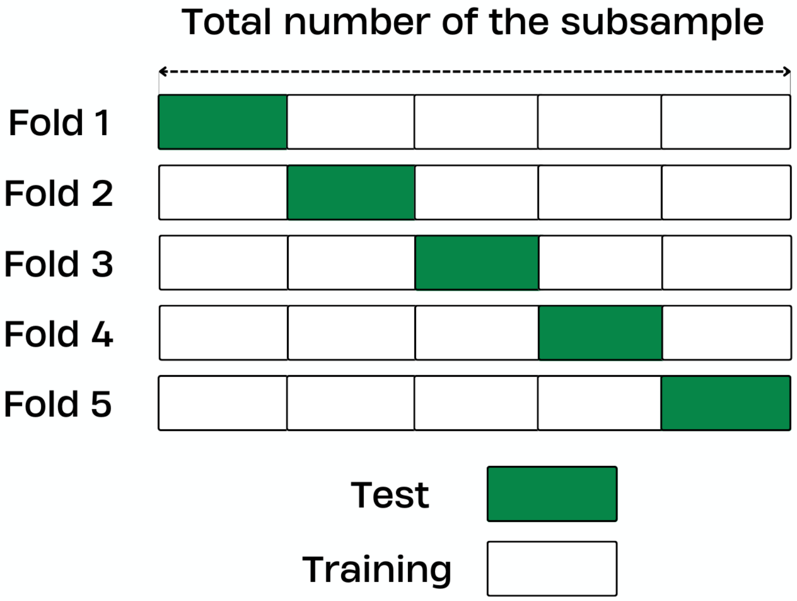
3.3 Cross-validation for Model Evaluation

Assessing the efficacy of a model is fundamental for comparison, allowing for the identification of the optimal model that best corresponds to the given data. This process involves dividing the dataset into a training set, which is utilized for tuning and educating the model, and a test set, acting as an unbiased platform for evaluating the model's performance. [Tibshirani & Friedman, 2001] have explored several strategies for model selection, highlighting cross-validation as a particularly prevalent method. They define the loss metric, which measures the discrepancy between Y and $\widehat{f}(X)$ using the training data, in the following manner:

$$L(Y, \widehat{f}(X)) = \begin{cases} (Y - \widehat{f}(X))^2 & \text{quadratic error} \\ |Y - \widehat{f}(X)| & \text{absolute error,} \end{cases} \tag{3.24}$$

Cross-validation evaluates the predictive accuracy on unseen data, quantified as $Err = E[L(Y, \widehat{f}(X))]$, within a specific training dataset $\mathcal{T}$. The K-fold cross-validation, a widely adopted approach, involves partitioning the entire training dataset into K unique subsets and alternating between training and testing roles for these subsets. For instance, with $K = 5$, as depicted in Figure 3.3, the dataset is divided into 5 subsamples, each split into training and testing groups. As shown in the figure, the division ratio for both the initial dataset segmentation and at this stage commonly adheres to a 70/30 or 80/20 split. However, researchers may opt for different proportions based on their preferences.

Figure 3.3: Split structure for K-fold cross-validation for K=5.



Source: author.

Consider the function $k : \{1, \ldots, N\} \mapsto \{1, \ldots, K\}$ that maps each observation i to one of K partitions via random distribution, where $\widehat{f}^{-k}(x)$ represents the model fitted on the dataset excluding the k-th subset. The evaluation of the prediction error through cross-validation is thus defined as follows:

$$CV(\widehat{f}) = \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \widehat{f}^{-k(i)}\left(x_i\right)\right) \tag{3.25}$$

In most cases, the choice of K is either 5 or 10. For the group of models $f(x, \alpha)$, involving a parameter that can be optimized, denoted as α, we have $\widehat{f}^{-k}(x, \alpha)$ as the α-th model fitted to the k-th segment of omitted data. Consequently, for this collection of models, the aim is to determine $\widehat{\alpha}$ that leads to the minimum:

$$CV(\widehat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \widehat{f}^{-k(i)}\left(x_i, \alpha\right)\right) \tag{3.26}$$

Thus, cross-validation serves to identify the best parameter or hyperparameter settings, optimizing the model's fit to the data at hand. The standard steps for conducting K-fold cross-validation are described as follows:

1. Split the dataset into K folds for cross-validation randomly.
2. For each fold $k = 1, 2, \ldots, K$:
   (a) Select a "promising" group of predictors that show a strong univariate correlation with the outcome variable, using data from all but the k^{th} fold.
   (b) Construct a multivariate model using only these selected predictors, excluding data from the k^{th} fold.
   (c) Use this model to predict the outcome for the data in the k^{th} fold. Aggregate the prediction errors from each fold to compute the overall cross-validation prediction error.

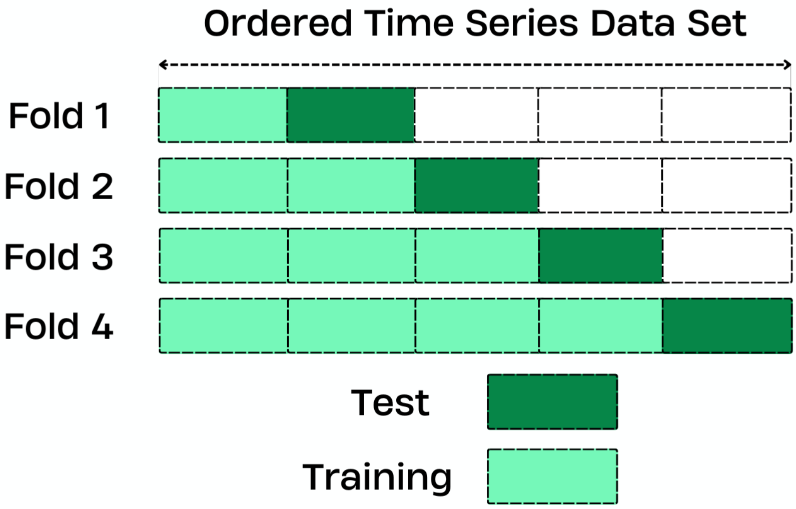
3.3.1 Cross-validation for Time Series

In time series forecasting, the goal is to leverage historical data to forecast future events. However, the application of random splits in K-fold Cross-Validation disregards the sequential nature of the data. Without any adjustment, this approach is unsuitable for time series analysis, as highlighted by [Medeiros et al., 2021] and [Athey & Imbens, 2019], due to its failure to acknowledge the temporal dependency within the dataset.

So the unique nature of time series data necessitates a specialized form of cross-validation called pseudo-out-of-sample evaluation, a common practice within the literature. This approach involves dividing the data into training and testing portions while maintaining the chronological sequence, a method exemplified by [Nosratabadi et al., 2020]. As detailed in Figure 3.4, the pseudo-out-of-sample technique ensures adherence to the time series' inherent order, preventing the use of future observations for past predictions by setting aside a segment of the dataset. The methodology, particularly illustrated through fold 3, is a standard practice in time series forecasting.

Although the common preference for pseudo-out-of-sample evaluation in time series forecasting with ML models, [Bergmeir et al., 2018] present compelling evidence that supports the effectiveness of cross-validation applied to traditional models approaches for time series prediction like autoregressive models. Their research suggests that K-fold cross-validation is particularly advantageous when the prediction errors are uncorrelated,

Figure 3.4: Pseudo out-of-sample evaluation split structure with K=4.



Source: Author

a situation often encountered with machine learning model applications. For a time series modeled by a nonlinear regression equation:

$$y_t = g(\mathbf{x}_t, \boldsymbol{\theta}) + \varepsilon_t \tag{3.27}$$

Where $g(\cdot)$ is a continuous, differentiable function about the parameter vector $\boldsymbol{\theta}$ for all inputs $\mathbf{x}_t = (y_{t-1}, y_{t-2}, ..., y_{t-p})'$, and ε_t represents the error term. Suppose $\{\tilde{y}_t\}_{t=1}^{m}$ denotes a sequence with the same distribution as the observed data $\{y_t\}_{t=1}^{n}$ (akin to future values of the series), with $\tilde{\mathbf{x}}_t = (\tilde{y}_{t-1}, \tilde{y}_{t-2}, ..., \tilde{y}_{t-p})'$.

The prediction error (PE) for this nonlinear setup is defined as $PE = E\left\{\tilde{y} - g(\tilde{\mathbf{x}}_t, \widehat{\boldsymbol{\theta}})\right\}^2$, with $\widehat{\boldsymbol{\theta}}$ obtained by minimizing the error function. Cross-validation estimates the PE using a training dataset comprising $\{(\mathbf{x}_j, y_j)\,; j = p+1, ..., n, j \neq t\}$ and a testing dataset of $\{(\mathbf{x}_t, y_t)\}$. This approach to estimating PE via cross-validation is described as follows:

$$\widehat{PE} = \frac{1}{n-p} \sum_{t=p+1}^{n} \left\{y - g\left(\mathbf{x}_t, \widehat{\boldsymbol{\theta}}_{-t}\right)\right\}^2, \tag{3.28}$$

Here, $\widehat{\boldsymbol{\theta}}_{-t}$ stands as the estimation of $\widehat{\boldsymbol{\theta}}$ for the training set, excluding the data available in the test set. To ensure the effectiveness of cross-validation, $\widehat{PE}$ should closely approximate PE. To achieve this, three prerequisites must be met:

1. First and foremost, the series being studied should be a stationary process;
2. $\widehat{\boldsymbol{\theta}}_{-t}$ should serve as a consistent estimator of $\boldsymbol{\theta}$.
3. The errors ε_t should adhere to martingale difference sequences, implying an absence of serial correlation within these errors. This assumption is crucial for the soundness of cross-validation.

When all three requirements are satisfied, these authors theoretically establish that $\widehat{PE} \xrightarrow{p}$ P E. The authors' empirical application further demonstrates that applying cross-validation for time series forecasting yielded comparatively more minor errors in contrast to employing pseudo-out-of-sample evaluation. This approach effectively tackles the issue of potential overfitting.

3.4 Hyperparameter Tuning

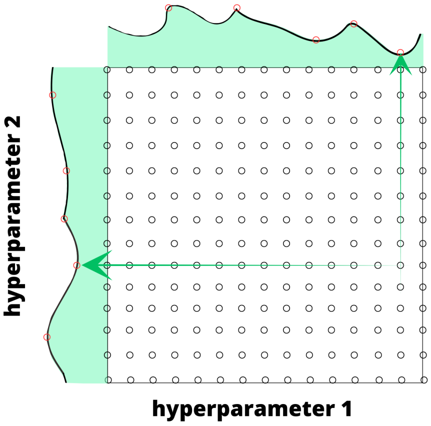
Generally, statistical learning methods contain one or more parameters, called hyperparameters, that penalize model complexity. Hyperparameters are a crucial part of many methods since it is the primary tool to address overfitting in complexity models. For example, the penalty term of the Elastic Net on λ is a type of hyperparameter. The λ controls the degree of the penalty imposed on the coefficients of a linear model.

Selecting values for one or more hyperparameters in statistical learning methods is generally called hyperparameter tuning, and there are many ways to perform it. The process of K-fold Cross-Validation is by far the most popular when performing hyperparameter tuning. We have already explained how this method works previously. To identify the hyperparameters that significantly enhance your model's performance, several strategies can be employed being the most common:

- Grid Search
- Randomized Search
- Bayesian Optimization
- Genetic Algorithms

Among these, Grid Search and Randomized Search are the most famous approaches and are considered "brute force" methods. This implies that hyperparameters are selected without leveraging prior information, relying instead on systematically testing various options to find a compelling combination. For this work, we decided to use Grid Search Hyperparameter Optimization. The process of grid search for two hyperparameters is illustrated in Figure 3.5. The optimal values for the hyperparameters are found at the intersection of the two green arrows. These values correspond to the lowest error statistic, and the green area on the graph shows its error distribution.

Figure 3.5: Error Statistics Distribution



Source: Adapted from [Bergstra & Bengio, 2012]

When optimizing through grid search, it's crucial to remember that the process can be time-consuming. All possible candidates are given equal weight during the hyperparameter mapping. In Table 4, you will find the machine learning models utilized along with their optimized hyperparameters. This study optimized the hyperparameters through the K-fold cross-validation method with K set to 10, using the training set and grid search technique.

Table 4: Hyperparameters Optimized in ML Models

Machine Learning Model	Hyperparameter
Elastic Net	Regularization parameter λ and α
Gradient Boosting	Number of Boosting Stages Learning rate Maximum depth of the Individual Regression
Random Forest	Number of Trees Maximum Depth of the Trees
Adaptive Boosting	Weight Applied to Each Regressor Loss Function Number of Estimators
XGBoost	Number of Trees Maximum Depth of the Trees Learning Rate

Source: author

Chapter 4

Methodology

Forecasting inflation in a data-rich environment in an emerging country like Brazil is challenging. As discussed earlier, machine learning techniques have revolutionized high-dimension inflation forecasting, offering numerous advantages. However, determining which variables are crucial can be difficult. Various features related to inflation forecasting are highlighted in the literature, and selecting the most significant ones can be complex. Therefore, following a process that enables us to identify the essential variables and discard the rest is advisable. To this end, we propose a workflow that involves a subset selection process before running our nonlinear models and analyzing the forecast results using Shapley values.

4.1 Overall Structure

In this study, we initiate our analysis by conducting comprehensive feature engineering, leveraging an extensive database with various features alongside their corresponding treatments. This initial phase is grounded in a thorough literature review, detailed explicitly in Chapter 2, where we explore contributions from multiple authors on inflation forecasting through Machine Learning (ML) techniques, with a particular focus on emerging markets. Subsequently, we undertake a target analysis to determine the most effective modeling strategies and treatments for our dependent variable, a critical step for linear models, such as the shrinkage method. Building upon this, we apply a feature selection (variable selection) process utilizing the Elastic Net shrinkage model technique, renowned for its robustness, with Recursive Feature Elimination (RFE) algorithm approach where it is fitting the model multiple times and in each iteration, remove the weakest features. The selected features during the feature selection step, determined by the shrinkage method, were then applied to non-linear models discussed in Chapter 3.

It is essential to highlight that we scale the predictors using Min-Max scaling (Normalization) to prevent the magnitude of the variables from affecting the models used in this study following [Lazzeri, 2020] and [Geron, 2019]. feature scaling is one of the most critical transformations you must apply to your data when using ML models. With few exceptions, Machine Learning models do not need to use any feature scaling technique. There are two common types of feature scaling methods According to [Geron, 2019]: Min-Max scaling (Normalization) [1] and Standardization (Z-score scaling) [2]. We applied Min-Max scaling following [Özgür & Akkoç, 2021] and [Wang et al., 2020].

During the non-linear model prediction, we forecast the IPCA headline and report four different accuracy metrics: mean absolute error (MAE); root mean squared error (RMSE); mean squared error and (MSE) for the out-of-sample period. We utilize the Model Confidence Set (MCS) test, as developed by [Hansen et al., 2011], to assess the accuracy of our machine learning methods and identify the most effective non-linear model for our analysis. The MCS test identifies a Superior Set of Models (SSM) that includes the best-performing model with a confidence level of $100(1-\alpha)\%$, allowing for adjustable selection criteria via α. This method involves a series of hypothesis tests to eliminate the lowest-performing models based on their predictive capacity, using a specific loss function. Ultimately, this process results in a set of top models at a predetermined confidence level, offering a more comprehensive comparison than the *Diebold-Mariano test* test's pairwise approach.

Throughout the hyperparameter optimization and train and test step, we set the code to use routine to refit and increase training size utilizing the dataset [3]. In other words, we use the time series backtesting with refit. According to [Skforecast, 2023] in this approach, the model is trained before making predictions each time, and all available data up to that point is used in the training process. Image 4.1 shows how this approach works.

In a nutshell, our research approach is systematically outlined as follows:

1. Feature Engineering: create a comprehensive database of features and their corresponding treatments based on information gathered from literature;

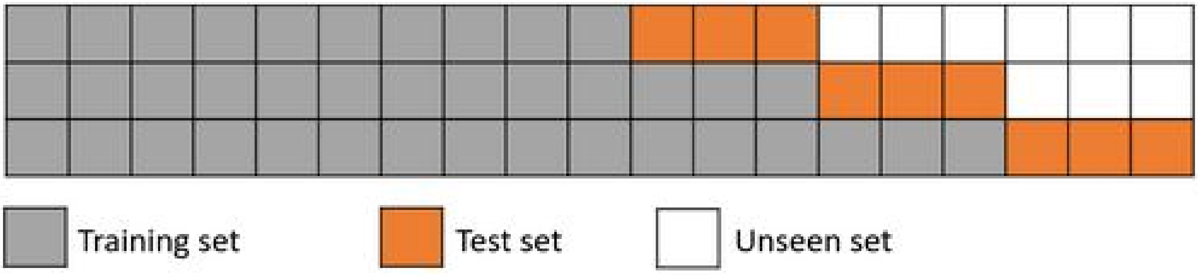
2. Target Analysis: Execution of target analysis to ascertain the most effective modeling approach;

[1] According to [Geron, 2019], Min-max scaling or normalization is quite simple: values are shifted and rescaled so that they end up ranging from 0 to 1. We do this by subtracting the min value and dividing by the max minus the min.

[2] According to [Geron, 2019], standardization operates differently: it begins by subtracting the mean value, ensuring that standardized values consistently possess a mean of zero, and subsequently divides by the standard deviation, resulting in a distribution with unit variance. Unlike min-max scaling, standardization doesn't confine values to a specific range. This characteristic might challenge certain algorithms (e.g., neural networks). Nevertheless, standardization is considerably less influenced by outliers.

[3] To perform it, we set the parameter $refit = True$ in *Skforecast* and *Sklearn* grid search for example.

Figure 4.1: Backtesting with refit and increasing training size (fixed origin).



Source: [Skforecast, 2023].

3. Feature Selection: Selection of features based on shrinkage model techniques, specifically employing the Elastic Net and using the Recursive Feature Elimination (RFE) approach;

4. Non-Linear Forecast: Application of the selected features in nonlinear models for inflation prediction;

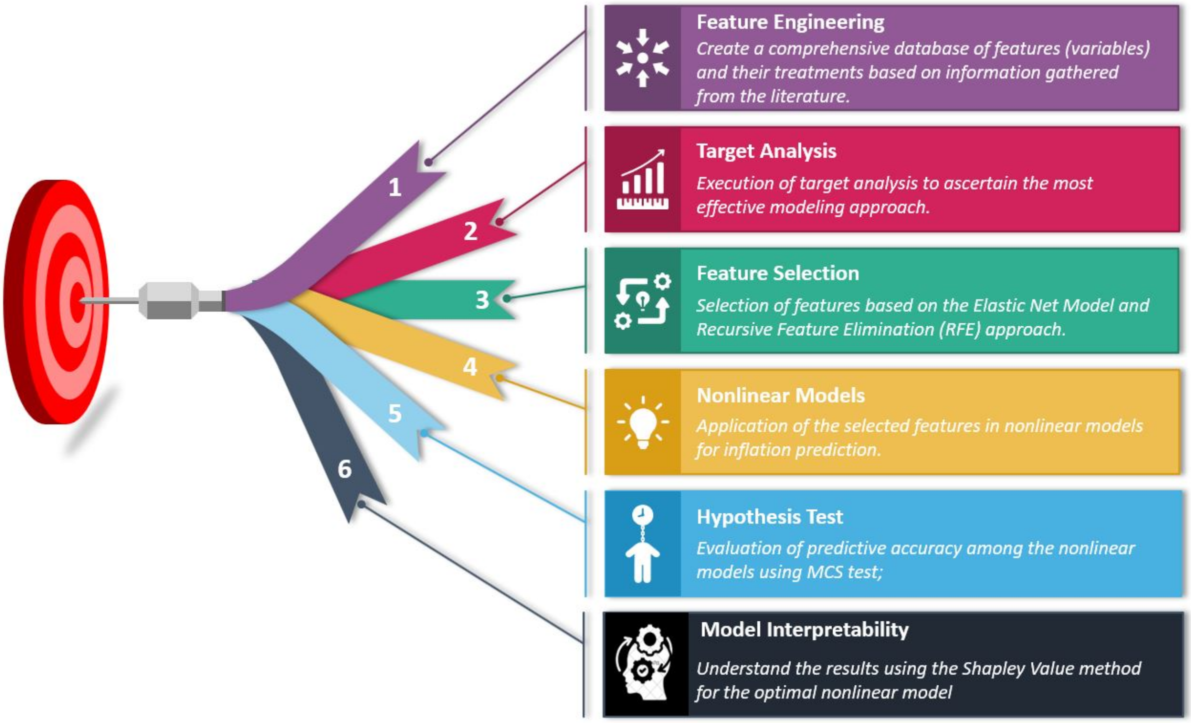
5. Hypothesis Test: Evaluation of predictive accuracy among the nonlinear models using MCS test;

6. Model Interpretability: Understand the results using the Shapely Value method for the optimal nonlinear model.

Figure 4.2 shows how the process works in our method:

In this study, we employ the R and Python programming languages, along with their respective packages, for all model analyses and data processing tasks. Specific languages and packages are referenced in each section, aligned with their application.

Figure 4.2: Research Approach


1
2
3
4
5
6
Feature Engineering
Create a comprehensive database of features (variables) and their treatments based on information gathered from the literature.
Target Analysis
Execution of target analysis to ascertain the most effective modeling approach.
Feature Selection
Selection of features based on the Elastic Net Model and Recursive Feature Elimination (RFE) approach.
Nonlinear Models
Application of the selected features in nonlinear models for inflation prediction.
Hypothesis Test
Evaluation of predictive accuracy among the nonlinear models using MCS test;
Model Interpretability
Understand the results using the Shapley Value method for the optimal nonlinear model

Source: Author

Chapter 5

Empirical Analysis

This chapter will demonstrate the outcomes of implementing the models discussed earlier. We will also assess their effectiveness using forecast error statistics and compare their performance. First, we will perform a target exploratory analysis after outlining the database and its treatment, from which we will use to perfom a feature selection step using LASSO method. Finally, we will present our comparative analysis for nonlinear models.

The entirety of our ETL process and tests related to trend, seasonality, structural breaks, and stationarity treatment procedures were implemented in R, utilizing packages such as tidyverse, adf.test, tseries, moments, tsfeatures, funtimes, strucchange, and urca. To run the models used in this study (Shrinkage and Nonlinear Models), we use Python and its packages, sklearn and skforecast.

5.1 Target Exploratory Analysis

Before conducting a forecasting analysis, it is crucial to inspect our target variable. Let's perform some exploratory data analysis (EDA) or, more specifically, a time series analysis. The purpose of doing that is to gain insights related to the target, which in this case is the monthly Brazilian Consumer Price Index (IPCA), essentially a time series.

At this moment, it's important to distinguish between time series analysis and forecasting. According to [Lazzeri, 2020], while these two domains are closely intertwined, they serve distinct purposes: time series analysis focuses on uncovering the inherent structure and extrapolating concealed patterns within your time series data to extract valuable insights, such as trends or seasonal variations. Scientists typically employ time series analysis for the following reasons:

1. To gain clear insights into the underlying structures of historical time series data.

2. To enhance the quality of interpreting time series features, providing more informed insights into the problem domain.

3. To preprocess and conduct high-quality feature engineering, resulting in a more comprehensive and profound historical dataset.

Still, according to [Hyndman & Athanasopoulos, 2021], conducting EDA before any forecasting exercise is crucial to yield good results. Then, let's dive into this in the following sections to identify seasonality, trend, and structural breaks in our target.

5.1.1 Seasonality and Trend Analysis

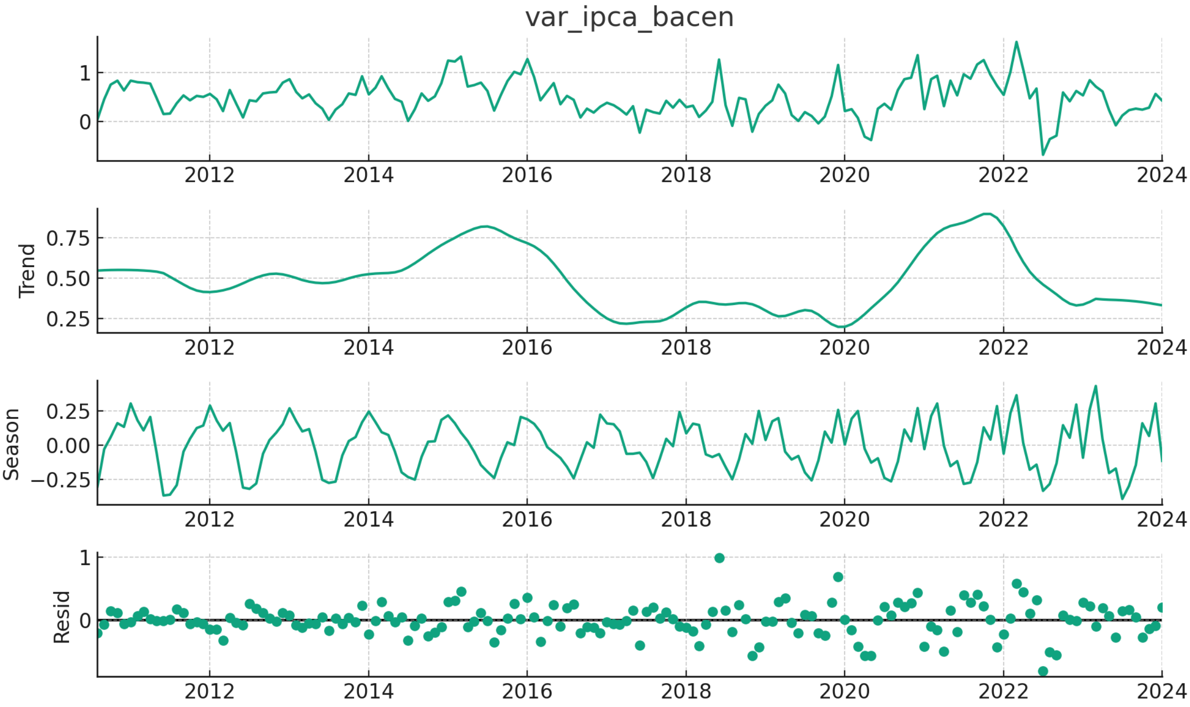
As first step, we applied a method to identify seasonality and trend on the target. Time series decomposition is a technique that allows an understanding of the different patterns present in the series. It identifies and isolates the time series into distinct components and is an important tool for exploratory analysis. Let y_t be a time series with n observations, the additive decomposition considers that the time series is the result of the combined effect of three components, namely, $y_t = S_t + T_t + R_t$, where S_t, T_t, and R_t are the seasonality, trend and residual components respectively.

The Seasonality and Trend decomposition using Loess STL method stands out among decomposition methodologies for its effectiveness and was therefore selected for application in this study. This method, as proposed by Wang and further elaborated by [Wang et al., 2006], allows for the precise calculation of trend and seasonality components. The STL decomposition of the time series data reveals the trend, seasonal, and residual components, with the seasonal component displayed in one of the subplots. Figure 5.1 illustrates the STL decomposition of the target time series. This visual tool is crucial for evaluating the relative influence of each component. Using this tool is possible to see a trend in some moments and a clear seasonal tendency on time series.

In our analysis, we utilized the Mann-Kendall test as a supplementary method to the STL test for detecting monotonic trends, either positive or negative, within the dataset. As delineated by [Zach, 2020], the test's null hypothesis posits the absence of any trend, while the alternative hypothesis suggests the presence of either a positive or negative trend. This non-parametric approach is advantageous for identifying trends in time series data without conforming to a specific distribution model. The result of the Mann-Kendall Trend test, indicated by a p-value of 0.02, substantiates the existence of a statistically significant trend in the data under study, given the p-value falls below the conventional threshold of 0.05.

Developing an appropriate strategy for addressing the observed trends and seasonality within our target variable is crucial. Although some studies, such as that by

Figure 5.1: Seasonality and Trend decomposition using Loess (STL) test



Source: author.

[Ouyang et al., 2021], suggest that incorporating such strategies in machine learning models does not necessarily enhance predictive performance, our research opts to implement target lag features as a method to navigate these issues in non-linear models. In contrast, we adjusted our target seasonally using a moving average approach for linear models in feature selection.

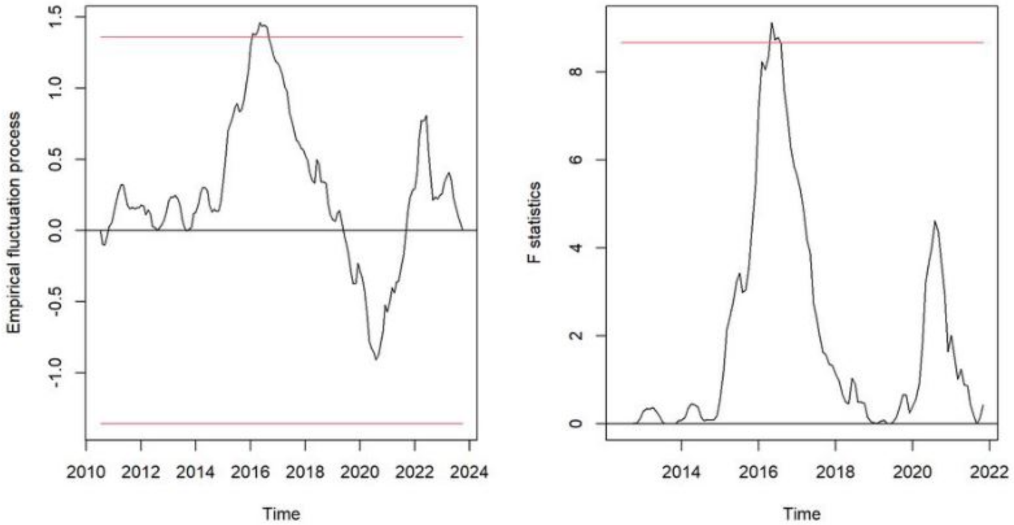
5.1.2 Structural Breaks Analysis

Detecting structural changes in time series is crucial in econometrics. This study applied three structural break tests: the CUMSUM-OLS test, a test based on F-statistics, and the Bai and Perron test. The CUMSUM-OLS test uses cumulative sums of standardized residuals to evaluate the stability of coefficients in a linear regression model. Under the null hypothesis of parameter stability, values outside a specified range suggest structural changes over time. The F-statistics-based test (Chow test) applies tests sequentially to examine the stability of breaks in linear models. It calculates an F-statistic for each potential breakpoint within a range, rejecting the null hypothesis of structural stability if any statistic exceeds a critical value. The Bai and Perron method was used to estimate multiple structural break points. The method seeks to identify m breakpoints, which is equivalent to $m + 1$ segments. The segments are chosen to minimize the residual sum of squares (RSS) and the Bayesian information criterion (BIC) [Kleiber & Zeileis, 2008].

Regarding the results found, the CUMSUM-OLS test yielded a p-value of 2.7% (0.0278), below the 5.0% significance level, indicating that model parameters are not stable throughout the evaluated period. This suggests the presence of a structural break in the series. Figure 5.2 illustrates the results of the CUMSUM-OLS test on the left, including the 5.0% significance limits. The OLS-based CUSUM process exceeded its limit, providing evidence of a structural change in the series in 2016, reaching its peak.

The test based on the F statistic (Chow test) also calculated the F value for potential structural breakpoints. The right chart in Figure 5.2 displays the F statistic process and its IPCA monthly percentage change limit. Notably, the process exceeded the limit, and both threshold crossings and p-values (p = 0.040) below the 5.0% significance level suggested a departure from the null hypothesis of structural stability. This further indicates the presence of a structural break. The test pinpointed May 2016 as the structural break point in the value of the IPCA time series differentiated index number, corresponding to when the process reached its maximum value.

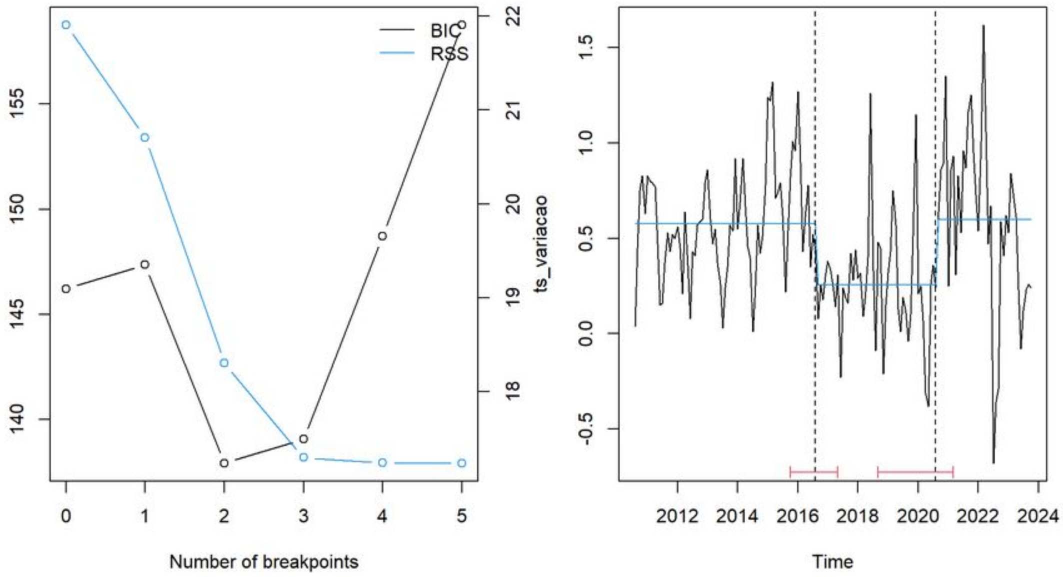
Figure 5.2: CUMSUM-OLS and F-statistic tests



Note:
The CUMSUM-OLS chart is on the left and the F-statistic tests chart are on the right.

The Bai and Perron test evaluated different potential structural breakpoints, and the selection was made by minimizing the BIC and RSS. The values for each number of structural breakpoints are shown in 5.3. It is noted that the BIC reaches its minimum value at m=2. Therefore, the test was applied with 2 potential structural breakpoints. The structural breakpoints identified by the Bai and Perron method in the IPCA monthly percentage change were in August 2016 and August 2020. The IPCA time series monthly percentage variation, adjustments made, and confidence intervals for the structural breakpoints are shown in Figure 5.3 (right).

Figure 5.3: BIC and RSS - Bai and Perron Test



The results of the three tests suggest the existence of a structural break in the target but differ in terms of the dates of occurrence. The three tests identified a structural break in 2016, in May for the CUMSUM tests and F test, and in August for the Bai and Perron tests. This suggests that, on average, the monthly percentage changes in the IPCA from 2010 to 2016 differ from 2017 to 2023, with a structural break point in May or August 2016. Furthermore, the Bai and Perron test also identified a structural break in August 2020. This suggests that, on average, the monthly percentage changes in the IPCA during the period from August 2010 to July 2020 are different compared to the period from September 2020.

Given the results of the structural breaks tests, we created a dummy to signal to the linear models (Elastic Net) applied in feature selection used in this work that the target's level changed between May 2016 and September 2020. For non-linear models, we did not apply this dummy.

5.2 Time Series and Transformations

We used 74 macroeconomic time series as columns (features) with 162 observations to predict the methods and models discussed in previous chapters. The data spans from August 2010 to January 2024 as rows (objects). The choice of feature and their transformation were based on [Araujo & Gaglianone, 2023] and [Garcia et al., 2017], the variables are listed in Table 5.

We followed our literature and applied one of six transformations to each variable, including (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $\ln(x_t)$; (5) $\Delta \ln(x_t)$; (6) $\Delta^2 \ln(x_t)$. Table 5 details the variables' names, sources where the time series can be found, treatments to turn them stationary, and original units. It's crucial to emphasize that we conducted stationary tests even after adjusting the time series according to the reference, selecting only those time series that passed the tests, as you can see in Section 5.2.1.

In this study, we assume that the dataset is balanced, meaning that we eliminated missing values. Besides, we adopted a lagged until $s = 4$ periods following [Araujo & Gaglianone, 2023] to capture inertial inflation [1]. There is another reason to adopt a lag structure. In the real world, we can only use information from period t to predict what will occur in period $t + 1$. So, unfortunately, we do not possess adequate information for numerous variables regarding the intended month. It happens especially when predictors are released at the same time that our target, for example, some items of the IPCA market basket - IPCA tradable, IPCA nontradable, IPCA services, etc. So, we must use these predictors lagged, including the series we are attempting to forecast. Finally, we obtain a final object-feature matrix with a database containing 286 series without stationary tests being applied.

Table 5: Macroeconomic Variables

Description	Unit	Δ	Source
Inflation			
IPCA (consumer price index, headline)	% p.m.	1	Bacen
IPCA (consumer price index, tradables)	% p.m.	1	Bacen
IPCA (consumer price index, administered prices)	% p.m.	1	Bacen
IPCA (consumer price index, market prices)	% p.m.	1	Bacen
IPCA (consumer price index, no nontradables)	% p.m.	1	Bacen
IPCA (consumer price index, services)	% p.m.	1	Bacen
IPCA (consumer price index, industrial goods)	% p.m.	1	Bacen
IPCA (consumer price index, food at home)	% p.m.	1	Bacen
IPCA (consumer price index, semi-durable goods)	% p.m.	1	Bacen
IPCA (consumer price index, durable goods)	% p.m.	1	Bacen
IPCA (consumer price index, nondurable goods)	% p.m.	1	Bacen
Core IPCA - Exclusion Food and Energy (EXFE)	% p.m.	1	Bacen

[1] According to [Araujo & Gaglianone, 2023], inflation models usually comprise a rich lag structure, particularly in emerging countries, more prone to inflation inertia. Such a design should capture the dynamic relationship between inflation, past inflation, and key macroeconomic variables.

Description	Unit	Δ	Source
Core IPCA - Exclusion EX0	% p.m.	1	Bacen
Core IPCA - Exclusion EX1	% p.m.	1	Bacen
Core IPCA - Exclusion EX2	% p.m.	1	Bacen
Core IPCA - Exclusion EX3	% p.m.	1	Bacen
Core IPCA - Double Weight	% p.m.	1	Bacen
Core IPCA - Trimmed mean smoothed	% p.m.	1	Bacen
Core IPCA - Trimmed mean	% p.m.	1	Bacen
IPCA diffusion index	%	1	Bacen
IPCA-15 (consumer price index-extended 15)	% p.m.	1	Bacen
IPC FIPE (consumer price index)	% p.m.	1	Bacen
Core IPC-Br (core inflation)	% p.m.	1	FGV
IPC-Br (consumer price index)	% p.m.	1	FGV
IGP-M (general price index)	% p.m.	1	FGV
IGP-DI (general price index)	% p.m.	1	FGV
INCC (national index of building costs)	% p.m.	1	FGV
IGP-10 (general price index)	% p.m.	1	FGV
IPCA Monitor - headline	% p.m.	1	IBRE/FGV
IPCA Monitor - food and drinks	% p.m.	1	IBRE/FGV
IPCA Monitor - food at home	% p.m.	1	IBRE/FGV
IPCA Monitor - food away from home	% p.m.	1	IBRE/FGV
IPCA Monitor - shelter	% p.m.	1	IBRE/FGV
IPCA Monitor - residence items	% p.m.	1	IBRE/FGV
IPCA Monitor - clothing	% p.m.	1	IBRE/FGV
IPCA Monitor - transport	% p.m.	1	IBRE/FGV
IPCA Monitor - health and personal care	% p.m.	1	IBRE/FGV
IPCA Monitor - personal expenses	% p.m.	1	IBRE/FGV
IPCA Monitor - education	% p.m.	1	IBRE/FGV
IPCA Monitor - communication	% p.m.	1	IBRE/FGV
FX, Risk and Banking			
FX-rate (nominal exchange rate, R$ /US$)	Units	5	Bacen
U.S. dollar index (DXY)	Index	5	Bacen
Ibovespa	Index	5	Bacen
VIX CBOE	Index	5	Bloomberg
MSCI emerging countries	Index	5	Bloomberg
CDS (Credit Default Swap) Brazil 5 years	Basis Point	5	Bloomberg
Commodities			

Description	Unit	Δ	Source
CRB all commodities	Index	5	Bloomberg
CRB Food	Index	5	Bloomberg
CRB Metal	Index	5	Bloomberg
Oil Brent	US$/barrel	5	Bloomberg
Oil WTI	US$/barrel	5	Bloomberg
Baltic exchange dry	Index	5	Bloomberg
Monetary aggregates and Interest rates			
Monetary base	R$ thousand	5	Bacen
Nominal policy interest rate (Selic)	% p.a.	2	Bacen
U.S. Treasury 3 months nominal yield	% p.a.	2	Bloomberg
U.S. Treasury 2 years nominal yield	% p.a.	2	Bloomberg
U.S. Treasury 10 years nominal yield	% p.a.	2	Bloomberg
Consumption and Economic Activity			
IBC-BR	Index	5	Bacen
GDP (12-months accumulated, current prices)	R$/million	6	Bacen
Consumer confidence - Fecomercio	index	2	Bacen
Capacity utilization (manufacturing industry)	%	2	FGV
Electric energy consumption (commercial)	GWh	5	Bacen
Electric energy consumption (residential)	GWh	5	Bacen
Electric energy consumption (industrial)	GWh	5	Bacen
Electric energy consumption (other)	GWh	5	Bacen
Employment and income			
Unemployment rate (open)	%	3	IBGE
Formal employment created - South	Units	2	MTb
Formal employment created - Southeast	Units	2	MTb
Formal employment created - North	Units	2	MTb
Formal employment created - Northeast	Units	2	MTb
Formal employment created - Central-West	Units	2	MTb
Climate			
El Niño - Oceanic Niño Index (ONI)	Index	2	Bloomberg
Forecast			
IPCA Focus	% p.m.	1	Bacen

5.2.1 Stationarity tests

As highlighted by Granger (1974) [Granger & Newbold, 1974], the presence of unit roots in time series data can lead to spurious inferences, potentially creating false correlations between independent variables. Conversely, when time series data lacks sufficient differencing, essential information about the relationships between variables may be lost, resulting in increased estimator variances.

Ensuring the stationarity of time series data is crucial for accurate modeling and inference. It safeguards against spurious correlations and prevents the loss of vital information regarding variable relationships. Therefore, transforming non-stationary time series into stationary ones is an important step when working with such data.

For this purpose, we conducted a battery of stationary tests. Our methodology involved subjecting each time series to three different tests: the Augmented Dickey-Fuller (ADF) test, the Phillips-Perron (PP) test, and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. We employed a stringent significance level of 1% (0.01) to establish stationarity. Only those time series that passed all three tests simultaneously were retained in our dataset.

We documented the p-values obtained from these tests in a tabular format detailed in Appendix A, and only time series with statistically significant p-values were considered for further analysis. It is essential to note that the interpretation of these tests varies: a low p-value, typically below 0.01, suggests stationarity for the ADF and PP tests, whereas for the KPSS test, a low p-value indicates non-stationarity.

This rigorous approach to ensuring stationarity not only underscores the robustness of our analysis but also bolsters the reliability of subsequent modeling and inference in our research. Furthermore, we have reduced our sample from 286 time series to 156 variables. However, this number of variables does not account for any target lag that will be used when forecasting with the non-linear models. Instead, it includes a dummy variable to signalize a structural break

5.3 Shrinkage Method for Feature Selection

Basic methods for choosing essential parts of data (features) look at each feature's qualities and how they relate. Methods like variance thresholding and pairwise feature selection are popular, especially in finance and economics. They work by eliminating features that don't change much or are too similar to each other. However, a more straightforward way is to pick features based on how much they help a model perform better. In this sense, we performed a feature selection over the 156 stationary variables using shrinkage methods.

As we discussed previously, these approaches promote sparsity and guard against overfitting by accentuating the significance of a select subset of features. As such, they prove invaluable in machine learning, particularly when confronted with high-dimensional datasets or singling out the most relevant attributes. In practical applications, feature selection using Shrinkage Methods involves three steps:

1. Feature Scaling: Since Shrinkage Methods are sensitive to the scale of variables, it's common practice to normalize or standardize the data before applying any of this methods.

2. Hyperparameter Tuning: Techniques like k-fold cross-validation are used to find the optimal λ and α, which determines the number of features selected.

3. Interpretation: The features with non-zero coefficients after applying the choose methods are considered essential. However, the interpretability depends on the context and understanding of the domain.

So, the idea of using Shrinkage Methods for feature selection purposes is straightforward: we fit the method on a scaled version of our dataset and consider only features with a coefficient different from zero regarding the sparsity effect. We choose to work with Elastic Net because of its sparsity ability to deal with many correlated features, while LASSO needs a condition to work well with a highly correlated dataset.

As a first step in using shrinkage methods for feature selection, we must preprocess our features using a feature scaling approach. According to [Geron, 2019] for ML models, proceeding with the preprocessing feature scaling is crucial. Feature scaling ensures that all input features have similar scales, which can help these models converge faster and perform better. As mentioned in methodology chapter we applied Min-Max scaling (Normalization) to our shrinkage model.

We choose to split our data for feature selection, with 80% of the sample data being used to train and the other 20% being our study's testing set. Besides, this work's section was built using the *Sklearn* library ([Pedregosa et al., 2011]) that contains classes and functions necessary to adapt regression models to time series forecasting.

5.3.1 Hyperparameter Tuning

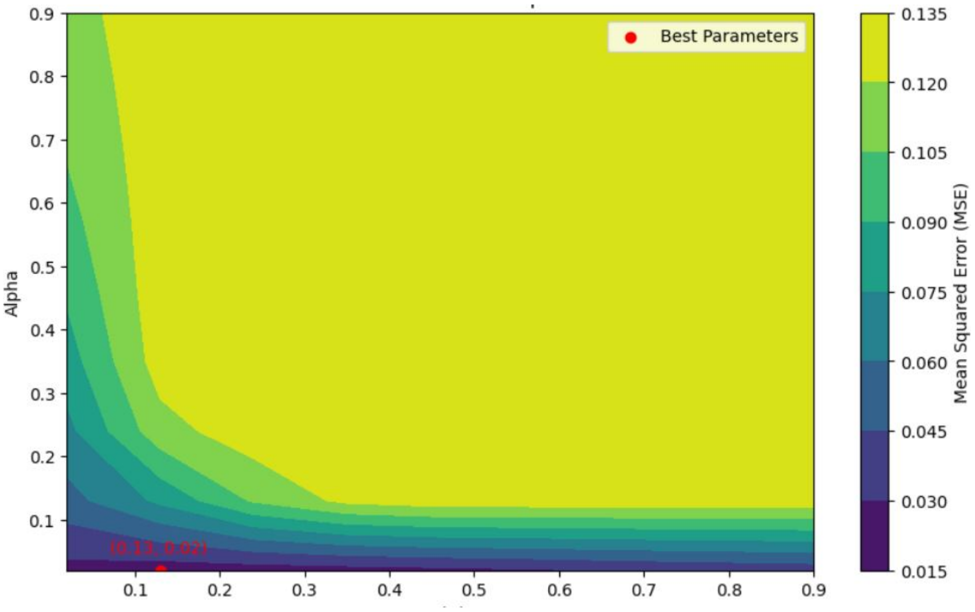
A crucial step in forecasting or feature selection when using shrinkage algorithms is to tune the hyperparameters. These hyperparameters are alpha (α) and lambda (λ) in the case of Elastic Net, and they play significant roles and are determined based on critical model evaluation metrics. Notably, α determines the algorithm type, while λ dictates the extent of the penalty applied during minimization. For instance, the optimal λ value, which results in the lowest error rate or information criteria, is considered the best choice. According to [Özgür & Akkoç, 2021], the optimal value of λ means the equilibrium point on the variance-bias trade-off.

Several criteria can be employed to select the optimal tuning parameters, including error-related metrics such as mean squared error (MSE), root mean squared error (RMSE), R2, and information criteria such as Akaike or Schwarz. In this study, we utilized a grid search combined with MSE to determine the optimal values of the tuning parameters empirically. Table 6 displays the optimal λ and α values for Elastic Net algorithms and chart 5.4 shows the evolution of MSE based on the parameter grid. The hyperparameters were optimized through the K-fold cross-validation method with K set to 5, using the training set and grid search technique.

Table 6: Best Hyperparameters

Model	Alpha (α)	Lambda (λ)
ElasticNet	0.02	0.13

Figure 5.4: Grid Search: MSE Evolution with Alpha and Lambda



5.3.2 Recursive Feature Elimination (RFE)

While shrinkage techniques such as ElasticNet are frequently used to select variables, it is important to note that the variables they select can vary over time and across different data samples. In other words, the number of variables chosen and the significance assigned to each (measured by their coefficients) can change when dealing with time series data. To address this variability and ensure the identification of consistently important variables across multiple interactions, we have opted for a methodology designed to pinpoint the most stable and influential features over time.

Sklearn [2] offers a tool called Recursive Feature Elimination (RFE) for this. RFE simplifies a model by iteratively removing the least important features until only the most useful ones remain. This backward feature elimination process involves initially fitting the model with all available features and then progressively eliminating the least significant ones, refitting the model each time until the desired number of features is reached [3]. RFE operates as a wrapper-style feature selection technique. Unlike filter-based methods, which individually score features and pick those with the highest (or lowest) scores, RFE wraps around a distinct machine learning algorithm, utilizing it at its core to aid in feature selection. Two essential configuration parameters merit attention in a RFE application: the number of features to be selected and the algorithm employed for feature selection. After selecting the best hyperparameters through optimization, we train our model using the RFE algorithm provided by *Sklearn* to identify the most frequent features over time.

In our study, we applied the RFE algorithm to a dataset of 162 observations. We employed a loop mechanism to apply RFE incrementally, starting with data from the first six months and adding one month at each iteration until all observations were included, totaling 156 iterations. During each iteration, the RFE algorithm processed the dataset, selecting the 30 most significant features through a step-wise elimination process guided by the ElasticNet model. This iterative approach allowed us to identify the most significant features across the entire dataset. Figure 5.5 illustrates the algorithm's application across the monthly datasets.

Based on their frequency across all 156 samples, the word cloud in figure 5.6 highlights the most relevant variables. According to our approach, the features most closely related to the target exhibit the highest significance level and frequency, as discussed in the next section.

[2] To call this function when using *scikit-learn* you need to use: feature_selection.RFE. There is an RFE algorithm specifically for Cross Validation sklearn.feature_selection.RFECV. We opted to work with the first option, just training the Elasticnet model without concern for performance, given that we will use the features in the non-linear models where performance is crucial.

[3] Which is set by the parameter n_features_to_select in sklearn.feature_selection.RFE.

Figure 5.5: RFE Loop

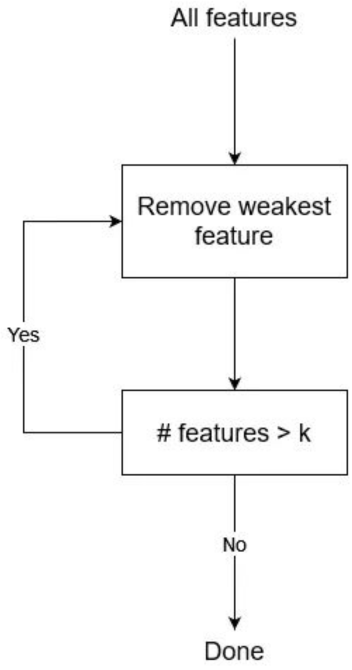
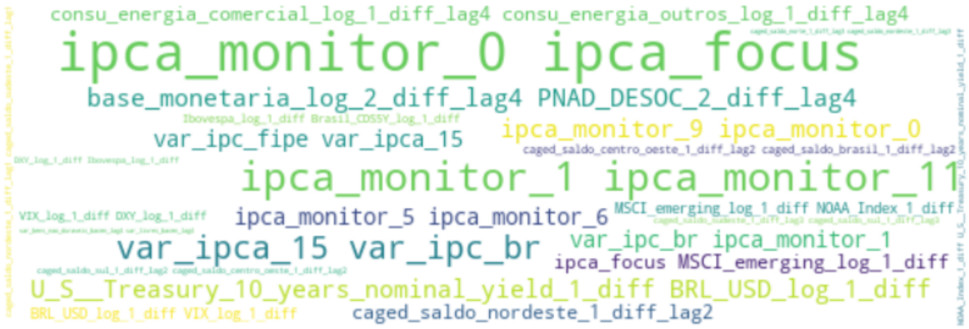


Figure 5.6: Word Cloud for features frequency.



5.3.3 Selected Features

After running the RFE algorithm and obtaining the most crucial feature through time, we selected the 30 most frequent to be used in our non-linear models. Their absolute and relative frequency are shown in appendix C, sorted by decreasing frequency. Chart 5.7 and appendix B shows their average coefficients through the samples. Twenty of them represent time series data associated with economic activity like the Fecomercio consumer confidence indicator [4] and energy consumption [5], Brazilian labor markets, like CAGED Payroll and unemployment rate (PNAD), climate (NOAA Index), or financial markets like nominal exchange rate (BRL/USD), DXY Index, and US Treasury 10 years. It's worth noting that all of these features exhibit different lag levels [6].

In our study variables related to economic activity, labor market, climate, or financial markets inherently carries a lag, even when the time series is contemporary to the target. For example, we have the contemporary indicator to the target for nominal exchange rate (BRL/USD), US Treasury for 10 years, and the DXY Index, but ElasticNet also selected their lagged values. Some features are not released contemporary to the target, like time series related to the labor market and energy consumption. These lagged features selected align with the observation that inflation models, particularly in emerging economies like Brazil, should incorporate a rich lag structure due to inflation inertia, as highlighted by [Araujo & Gaglianone, 2023]. Furthermore, specific economic indicators such as energy consumption and exchange rates have been previously identified by [Özgür & Akkoç, 2021] in their research on forecasting inflation in Turkey using Shrinkage Methods to perform a feature selection, demonstrating their relevance across different contexts.

Ten selected features are closely relate to the target variable or other inflation indicators in Brazil. Among them, subtypes of IPCA with different lags like IPCA - non-monitored prices (IPCA - Livres [7]) with lags of 1 and 4 periods, and IPCA-15 are significant. Additionally, there are other monthly inflation metrics in Brazil, such as IPC-Br and IPC-Fipe [8], released before our target, without any lag. These features are contemporary to the target and have reasonable importance in explaining the inflation dynamics in Brazil.

[4] consume_confidence_1_diff_lag3

[5] Energy consumption had six different metrics selected by our method. They are the second lag of residential energy consumption (consu_energia_residencia_log_1_diff_lag2), the first lag of commercial energy consumption (consu_energia_comercial_log_1_diff_lag1), the first and third lag of overall energy consumption (consu_energia_brasil_total_log_1_diff_lag1, consu_energia_brasil_total_log_1_diff_lag3) and third lag of other types of energy consumption (consu_energia_outros_log_1_diff_lag3).

[6] We applied up to 4 lags to these features, and you can find more details in Section 5.2.

[7] var_livres_bacen_lag1 and var_livres_bacen_lag4.

[8] var_ipc_br and var_ipc_fipe, respectively.

The most frequent predictor 'ipca_monitor_0', appearing almost 69,2% of all time and with the highest average coefficient, corresponds to a time series derived from the "Monitor da Inflação" [9], a database provided by IBRE/FGV, created as a previous proxy to the montly Headline IPCA. It aims to replicate the IPCA methodology, providing near-real-time estimates based on daily price changes [10]. Many features were created using "Monitor da Inflação", data that significantly influenced the model based on their coefficient levels. To distinguish them, we have incorporated the string "monitor" into their names[11]. These indicators are groups of IPCA [12], and their feature importance level reflects their weight on the Brazilian headline consume price index. For example, the monthly spending in the groups "Transportation" and " Food and Beverage" significantly impacted our target when considering their coefficient level.

The second most frequent variable, appearing in 59.6% of all samples, is a feature related to the Focus Readout ('ipca_focus')[13]. This variable is the median market expectation of our monthly target for every month. Given that this variable is a survey of the market for inflation ahead, it is unsurprising that it is highly relevant due to its frequent occurrence. However, according to our approach, this feature does not have the second-highest average coefficient; this position is held by IPC Fipe ('var_ipc_fipe'), another consumer price index released prior to our target (IPCA), as shown in chart 5.7. When considering the coefficient level, 'ipca_focus' has the fourth-highest value, not far from the second and third-highest values.

Indeed, considering the table in Appendix C, starting from the second position, it is possible to observe the difference between the most frequent features and their coefficient levels. From the third position onwards, the most frequent features are the nominal exchange rate, energy consumption, and unemployment rate in different lags, respectively[14]. Their average coefficients are in different positions and have different directional signs. A

[9] This service generates daily estimates of the IPCA, the country's official inflation index, which is the responsibility of IBGE. The calculation system also makes IPC-FGV results available with the same frequency. You can find more information at https://portalibre.fgv.br/monitor-da-inflacao

[10] We created a routine to turn the daily data into a monthly average for every feature that use data from "Monitor da Inflação" in our work.

[11] We generated a total of 12 indicators, yet only eight demonstrated stationarity in our tests, as detailed in Appendix A. Following the ElasticNet feature selection process, four indicators were chosen, namely: "ipca_monitor_0," "ipca_monitor_5," "ipca_monitor_1" and "ipca_monitor_11". The average coefficients associated with these indicators can be found in Appendix B.

[12] "Transportation" ("ipca_monitor_5"), "Food and Beverage group" ("ipca_monitor_1"), "Food at Home" ("ipca_monitor_11"), "Housing" or "Shelter" ("ipca_monitor_2"), "Education" ("ipca_monitor_8").

[13] According to BCB, the Focus Readout is a weekly survey of expectations that summarizes the statistics calculated over market expectations collected until the previous Friday. The report includes graphs and statistics regarding market expectations for price indices, economic activity, exchange rate, and Selic rate, among others. The expectations are provided by market analysts, not by the BCB itself.

[14] 'BRL_USD_log_1_diff' (37.2%), 'consu_energia_outros_log_1_diff_lag4' (36.5%), 'PNAD_DESOC_2_diff_lag1' (36.5%), 'PNAD_DESOC_2_diff_lag4' (35.9%)

notable case is the variable related to the unemployment rate, with three different lags selected by our method. However, the first and fourth lags are the most frequent and have average coefficients in different directions. While ’PNAD_DESOC_2_diff_lag1’ has a positive average coefficient, ’PNAD_DESOC_2_diff_lag4’ has a negative value. According to the literature that relates inflation and the labor market (unemployment rate in a Phillips curve, for example), they have an inverse relationship. In theory, we should only see negative coefficients for this variable, but this did not happen in our approach for the unemployment rate with one lag (’PNAD_DESOC_2_diff_lag1’); the other lags have a negative coefficient, aligning with the literature.

Still related to the signal for the variable coefficients, the majority are positive, as illustrated in Chart 5.7 and the appendix B shows their average values through all samples. However, eleven variables have a negative coefficient, suggesting that increasing one and keeping all others stable reduces the consumer inflation level. Nineteen features have a positive coefficient, suggesting that as the feature value increases, the target variable also increases. In other words, positive coefficients tend to impact inflation levels directly. For instance, the feature ipca_monitor_0 has an average coefficient of approximately 0.10, indicating a positive relationship with the target variable. If all other features remain constant, an increase in this variable is associated with an increase in the predicted value. Some of the features with a positive coefficient value are related to economic activity or the labor market in different lags, for example, energy consumption [15] and Caged Payroll [16]. A surge in these variables may indicate heightened aggregate demand within the Brazilian economy, potentially leading to increased domestic inflation.

It is important to point that some lagged targets were identified as significant predictors in our tests previously this study. These preliminary analysis indicates that incorporating lagged values of IPCA could enhance inflation forecasting accuracy. However, for the current stage of our study, we have opted to exclude these lagged features temporarily. They will be reintegrated in subsequent phases, particularly as we explore forecasting through non-linear models, where a broader array of lagged targets may prove beneficial.

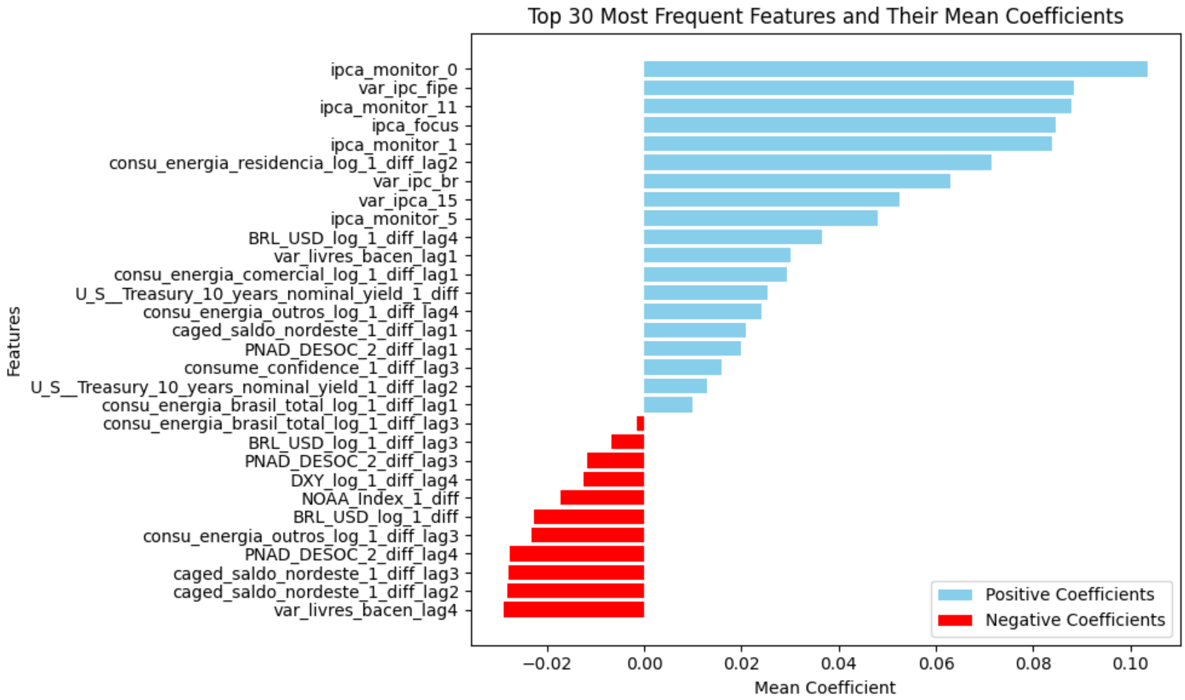
There is a curious result related to the Brazilian exchange rate [17] coefficient signal. This variable without lag (’BRL_USD_log_1_diff’) and with three lags (’BRL_USD_log_1_diff_lag3’ has a negative coefficient, but the same variable with four lags (’BRL_USD_log_1_diff_lag4’) have a positive coefficient. This situation suggests that domestic currency depreciation/appreciation correlates with higher/lower inflation rates in Brazil with different dy-

[15] consu_energia_residencia_log_1_diff_lag2,consu_energia_comercial_log_1_diff_lag1, consu_energia_outros_log_1_diff_lag4, consu_energia_brasil_total_log_1_diff_lag1 and consu_energia_brasil_total_log_1_diff_lag3

[16] caged_saldo_nordeste_1_diff_lag1

[17] Exchange rates (BRL/USD) reflect the relation of the domestic currency against the U.S. Dollar.

Figure 5.7: Coefficients Plot



Source: Author.

namics when the variable is dealing with lags or contemporaneous to the target. A positive coefficient reinforces a well-known correlation in the economic literature that can be attributed to their direct influence on the costs of imported goods and services. In other words, spending more money to buy imported goods and services will be necessary if the domestic currency depreciates and the exchange rate with four lags confirms it. On the other hand, the other selected features suggest that increases in nominal exchange rates may imply inflation reductions. This situation highlights the significance of exchange rates in driving domestic inflation and prompting proactive monetary policy measures.

5.4 Non Linear Forecast

After performing feature selection via a shrinkage method, we will advance our forecasting approach by applying non-linear models to the previously selected features. Decision tree approach models were chosen for their notable advantages, including interpretability, the capacity to manage unbalanced data, adeptness at handling missing values, and their non-parametric nature.

This work's section was built using the *Skforecast* library ([Amat Rodrigo & Escobar Ortiz, 2023]), which contains classes and functions necessary to adapt regression models from the *Scikit-Learn* library ([Pedregosa et al., 2011]) to time series forecasting. *Skforecast* allows us to create models with a given number of time series lags and, at the same time, work with exogenous variables in our models.

As highlighted in Section 5.3.3, reintroducing target lags into our analysis marks a pivotal juncture. The *Skforecast* package, treating target lag selection comparably to hyperparameter tuning[18], necessitates devising a lags grid to determine the most suitable lag choice for each predictive model. This approach has been systematically integrated into our hyperparameter tuning methodology using *Skforecast*. Regarding the sample split dataset, we apply the same approach as in the feature selection step, where the training set consisted of 80% of the time series and the other 20% of the test sample.

5.4.1 Hyperparameter Tuning

Such as in our feature selection step, the cross-validation technique for time series was used in the model training stage. This technique is a variation of k-fold cross validation, which allows maintaining the temporal order of data during the validation process as we explained in the section 3.3.1. In this procedure, the series is divided into several sequential subsets (folds), in our case is five, so that each subset is used as a test set once, while the previous sets are used as training sets. Thus, the size of the training set increases sequentially, and the model can be trained on increasingly more significant amounts of historical data, allowing a more accurate assessment of its predictive capabilities [Lazzeri, 2020]. The models' hyperparameters were selected using Grid Search, based on Time Series Split cross-validation. Table 7 shows the parameters chosen in each model.

The *Skforecast* package offers an indispensable tool for hyperparameter tuning by enabling immediate and automated backtesting with the optimal combination of lags and hyperparameters[19]. Essentially, the *Skforecast* library integrates a grid search strategy with backtesting, facilitating the identification of the most practical combination of lags and hyperparameters for superior prediction accuracy.

In the context of target lags, a decision was made to implement a parameter grid of 24 target lags across all models in this study. The hyperparameter tuning process with *Skforecast* involves a methodical examination of various hyperparameters and lags configurations to discover the optimal setup that yields the highest predictive performance.

[18] Additional information can be found at Skforecast Documentation.

[19] Upon setting the argument $return\_best = True$, the forecaster is retrained using all available data, employing the best combination of lags and hyperparameters [Skforecast, 2024].

Following the tuning process, the ideal number of lags and 30 selected variables during shrinkage approach were integrated into each non-linear model for the training (fit) and prediction phases (predict). Finally, the best hyperparameter set for each model was chosen using the mean absolute error (MSE).

Table 7: Hyperparameter Optimization Results - Non Linear Models

	Random Forest	Adaptive Boosting	Gradient Boosting	XG Boosting
Lags	2	6	10	2
Max Depth	5	-	3	10
N Estimators	100	150	200	150
Learning Rate	-	0.1	0.1	0.1
Loss	-	Exponential	-	-
MSE	0.0094	0.0087	0.0068	0.0079

Note 1: "N Estimators" the number of trees in the forest for Random Forest. For Others, specifies the number of weak models to use.
Note 2: "Learning Rate" defines the contribution of each weak model to the final prediction.
Note 3: "Lags" means the best amount of target lags features added to each model.
Note 4: "Max Depth" is the maximum depth of trees.
Source: Author.

5.4.2 Residual Analysis

Residual analysis is critical in any model prediction technique, whether machine learning or not. It examines the difference between predicted and observed values (residuals) to evaluate a model's performance and assumptions. Residual analysis assesses the discrepancies between expected outcomes and actual observations in a dataset. These residuals represent the model's errors and offer insights into its performance and potential areas of improvement. First, let's start with the visual inspection by plotting the residuals to check for any apparent patterns that might suggest they are not independently and identically distributed (I.I.D.) or lack of randomness. The I.I.D. assumption of residuals is crucial for the reliability of many statistical inference methods. This includes looking for autocorrelation (which would violate independence) and checking for consistent spread across the dataset to confirm they are identically distributed.

The charts 5.8 show how residuals behave for each ML model prediction. The residuals scatter around the zero line without forming any clear patterns or trends. This lack of pattern suggests a degree of randomness, which is suitable for the independence assumption. There does not appear to be a systematic change in the variance of residuals across the index the spread of residuals remains relatively consistent. This consistency supports the identically distributed assumption, but for a more comprehensive assessment, it is cru-

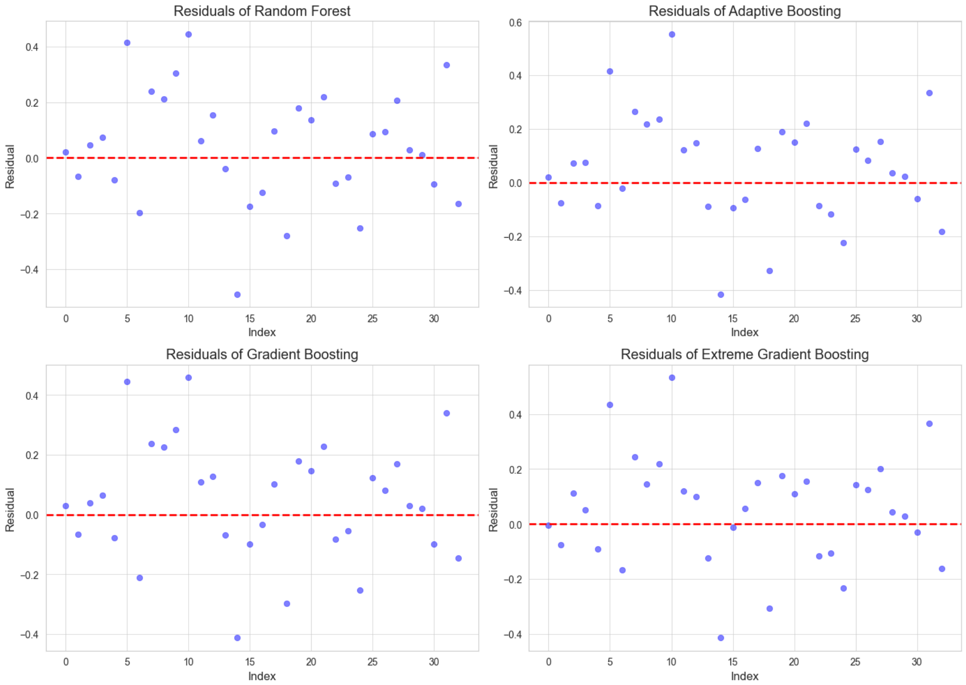


Figure 5.8: Scatter plot of ML models residuals.

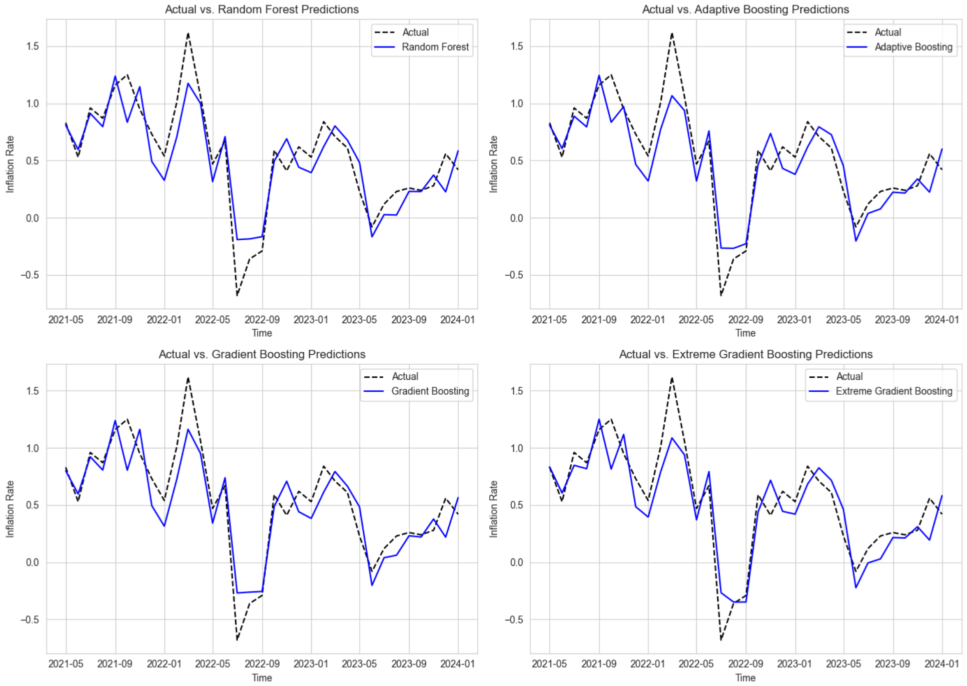
cial to perform formal tests. These tests can provide quantitative evidence to support or refute the visual insights. In this sense, we opted to apply the following statistical Tests: the Durbin-Watson test to check for autocorrelation in residuals; the Breusch-Pagan and White test for homoscedasticity; and the Shapiro-Wilk test to assess normality.

The Durbin-Watson statistic, which assesses autocorrelation in residuals, generally ranges from 0 to 4. Values around 2 imply a lack of autocorrelation, with figures below 2 indicating positive autocorrelation, and those above 2 suggesting negative autocorrelation. For the models assessed—Random Forest, Adaptive Boosting, Gradient Boosting, and Extreme Gradient Boosting—the statistics were 1.79, 1.65, 1.87, and 1.90, respectively. These results, being close to 2, suggest minimal autocorrelation across all models.

Concerning heteroscedasticity, the Breusch-Pagan and White tests yielded high p-values (greater than 0.05), indicating no significant evidence of heteroscedasticity within the residuals of our models. Similarly, the Shapiro-Wilk test for normality produced p-values well above the 0.05 threshold for all models, thereby supporting the hypothesis of residual normality.

In summary, the comprehensive testing of residuals—through visual examination and quantitative statistical tests—validates their reliability in representing independently and identically distributed data. The minimal autocorrelation, absence of heteroscedasticity, and adherence to normality across all models underline the robustness of their predictions. Therefore, the conducted residual analysis confirms that our machine learning models

Figure 5.9: Model's Actual vs. Predicted



meet the critical assumptions necessary for accurate predictive modeling.

5.4.3 Forecast Evaluation

Related to forecast results, first, we plot each model's actual vs. predicted inflation rates over time in image 5.9 and calculate the prediction errors to visualize their Box Plot in image 5.10. For chart 5.9, the closer the predicted values (blue line) are to the actual values (black dashed line), the better the model's forecasting accuracy.

Following this, the Box Plot of the residuals for each model in image 5.10 offers valuable insights into their prediction errors when forecasting Brazilian inflation rates. The medians of the residuals are relatively close to zero for all models, suggesting no significant systematic bias in the predictions. This is a positive indication of model performance. The variability of the residuals, as shown by the interquartile range (IQR), seems comparable across the models, with no single model displaying dramatically better or worse consistency in its predictions than the others. Some outliers are present for each model, indicating instances where the predictions were significantly inaccurate. This is common in predictive modeling, especially with complex phenomena like inflation. Overall, each model performs reasonably well, with no clear winner based solely on this analysis. However, factors such as the magnitude of errors, specific application needs, and computational efficiency might influence the choice of the best model for practical use.

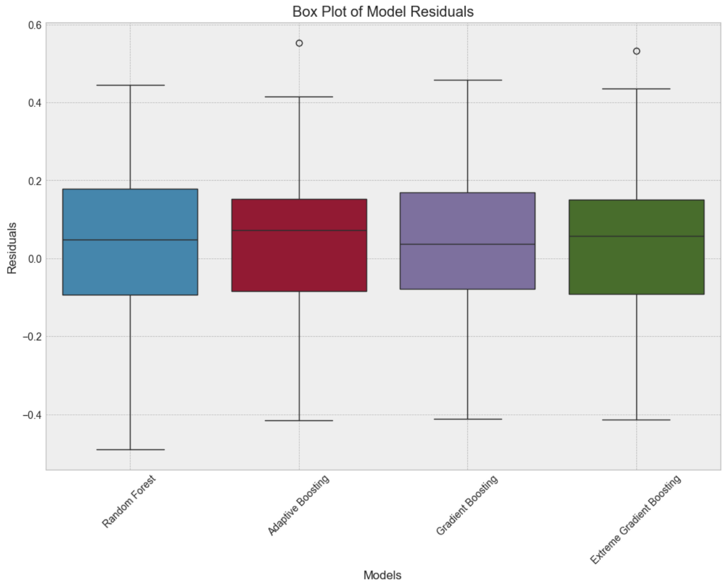


Figure 5.10: Box Plot Model Residuals

Related to the accuracy metrics, to assess the effectiveness of various models, we will rely on the following statistical measures[20]: mean absolute error (MAE), root mean squared error (RMSE), mean squared error (MSE). These are widely accepted metrics for evaluating how well a regression model performs when dealing with decision tree models. The definitions for each of these metrics are as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i| \tag{5.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}\,. \tag{5.2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.3}$$

where:

- y_i is the actual value,
- $\hat{y}_i$ is the forecasted value,
- n is the total number of observations.

In Table 8, a comparison is presented among the forecast statistics of machine learning models (out-of-sample).

[20] We choose not to use the R squared (R^2) as one of these metrics. According to [Spiess & Neumeyer, 2010], this accuracy metric can lead to wrong interpretations when choosing the best results among non-linear models.

Table 8: Model Performance Comparison (out-of-sample)

Model	MAE	MSE	RMSE
Random Forest	0.165	0.042	0.206
Adaptive Boosting	0.163	0.042	0.206
Gradient Boosting	0.161	0.041	0.202
Extreme Gradient Boosting	0.162	0.042	0.204

Note: The table values are rounded to three decimal places for clarity.

Regarding each model's performance, Gradient Boosting stands out with the lowest RMSE (0.202), MAE (0.161), and MSE (0.041) among the models tested. This indicates that it has the highest predictive accuracy. Therefore, it performs best in predicting inflation rates.

Extreme Gradient Boosting shows good performance, but its RMSE, MSE, and MAE are slightly higher than the Gradient Boosting model, suggesting it might not be the most precise model among those evaluated. Concerning Adaptive Boosting, it shows similar performance metrics to the Extreme Gradient Boosting model, but it has a slightly worse MAE and RMSE. Random Forest has the highest MAE, but its performance is still competitive when we consider RMSE and MSE being the same values for these metrics compared to Adaptive Boosting. Although the values are higher than the Gradient Boosting model.

In summary, Gradient Boosting is the most effective model for predicting Brazilian inflation, as indicated by its lower accuracy metrics. While all models perform reasonably well, the differences among RMSE, MAE, and MSE highlight the importance of choosing the suitable model based on the analysis's specific requirements.

5.4.4 Model Confidence Set (MCS)

To assess model performance using the evaluation sample defined initially, metrics provide an insight into out-of-sample test effectiveness. However, formal statistical tests remain imperative to determine the superior performance among the models forecasts. The prevailing method in literature is the *Diebold-Mariano Test* introduced by [Diebold & Mariano, 1995]. This test is commonly employed to gauge a model's predictive accuracy against a benchmark, assumes equal accuracy between models under its null hypothesis. However, when comparing forecasts from multiple models, a statistical technique capable of simultaneous comparison becomes necessary.

In this study, we utilize the Model Confidence Set (MCS) method, pioneered by [Hansen et al., 2011], to select the best forecasting models. Unlike the Diebold-Mariano Test, which requires a benchmark for comparison, MCS allows the simultaneous comparison of multiple models without needing a predefined benchmark. This method ranks models by their expected loss, determining superiority based on lower loss figures. The MCS technique identifies the most reliable models at a given confidence level. In other words, the MCS procedure yields a set of models that, with a certain confidence level, is assured to include the best models. The selection process uses user-defined loss functions, such as squared or absolute forecast errors, to compute a loss matrix and identify models with minimal expected loss. Through sequential testing, inferior models are excluded, allowing for a refined set of models based on MCS p-values, which users can apply significance levels to for final selection.

Given a set of models, $\mathcal{M}_0$, intended for comparison, we apply an 'equivalence test,' $\delta_{\mathcal{M}}$, and an 'elimination rule,' $e_{\mathcal{M}}$, to the set $\mathcal{M} = \mathcal{M}_0$. If $\delta_{\mathcal{M}}$ is rejected, it indicates that the models in $\mathcal{M}$ do not share the same predictive performance, prompting the use of $e_{\mathcal{M}}$ to eliminate models with comparatively poor performance. This process continues until $\delta_{\mathcal{M}}$ is no longer rejected, resulting in the final set $\mathcal{M}^*$, which comprises the model confidence set.

The comparison of models is conducted through the differences in their loss functions, denoted as $(d_{ij,t} \equiv L_{i,t} - L_{j,t})$, for models i, j within the set $\mathcal{M}_0$. The null hypothesis is defined as:

$$H_{0,\mathcal{M}} : E\left(d_{ij,t}\right) = 0, \forall i, j \in \mathcal{M} \subset \mathcal{M}_0.$$

The set of best models, $\mathcal{M}^*$, is identified as:

$$\mathcal{M}^* \equiv \left\{i \in \mathcal{M}_0 : E\left(d_{ij,t}\right) \leq 0, \forall j \in \mathcal{M}_0\right\}.$$

Algorithm for Model Confidence Set

The procedures for adopting the MCS are outlined in the following algorithm:

1. Set $\mathcal{M} = \mathcal{M}_0$.
2. Test for the hypotheses of equal predictive capacity (EPA) for the models in $\mathcal{M}$:
   (a) test for EPA–hypothesis: if EPA is not rejected define $\widehat{\mathcal{M}}^*_\alpha \equiv \mathcal{M}$ and report the $(1 - \alpha)$ confidence set.
   (b) test for EPA–hypothesis: if EPA is rejected:
      i. Define

$$d_i \equiv \frac{1}{m} \sum_{j \in \mathcal{M}} d_{ij}$$

where m is the number of models in $\mathcal{M}$. This statistic defines the performance of model i compared to the average of all other models.

ii. Determines the worst performing model in $\mathcal{M}$, defined as

$$i^* \equiv \arg\max_{i \in \mathcal{M}} \frac{d_{ij}}{\sqrt{\widehat{\operatorname{var}}\left(d_{ij}\right)}},$$

where $\widehat{\operatorname{var}}\left(d_{ij}\right)$ is the estimation of the variance of d_i. This maximized term is called the statistic t_{ij}.

iii. Remove model i^* from $\mathcal{M}$ and repeat step 2.

Our analysis employs squared forecast errors as loss functions, implemented in R with the *modelconf* package by [nielsaka, 2023]. The *estMCS* function is set to a 0.05 confidence level with 25,000 bootstrap resamples and employs the *Tmax* statistic, ensuring the method's robustness in model comparison and selection. The outcomes are delineated in Table 9. The greater the number of models eliminated, the more significant the model heterogeneity, but in our case, there is no model eliminated. According to our MCS test results, the most effective model was Gradient Boosting. The hierarchical structure shown in column "Rank" of these results is corroborated by analogous investigations employing accuracy metrics. Notably, lower MCS p-values suggest a diminished probability of a model being classified among the superior models.

Table 9: Set of Best Models for the series

Model	Rank	p-val	p-val MCS
Random Forest	2	0.66	0.66
Adaptive Boosting	3	0.60	0.66
Gradient Boosting	1	1.00	1.00
XG Boosting	4	0.49	0.66

Source: author.

Chapter 6

Model Interpretability

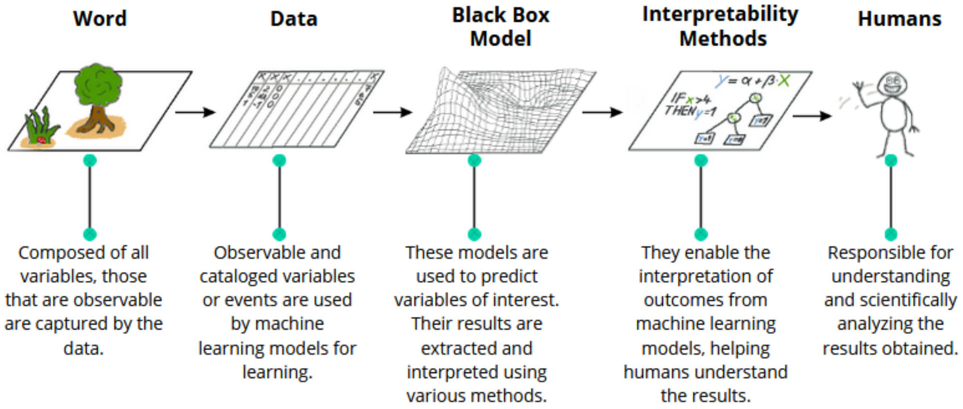
Previously, we learned about ML models and applied them to forecast our target. Some of these models are inherently interpretable as their parameters or structures can be easily understood. However, more complex models like random forests, gradient boosting, and XGBoost can be challenging to interpret, especially in settings with many predictors. This chapter will delve into the Shapley Value, a model-agnostic explainable artificial intelligence (XAI) method that can be applied to any machine learning model. Model-agnostic XAI interpretability methods provide vital insights into the models' mechanisms for making predictions, even when complex.

6.1 Why Does Interpretability Matter?

Machine learning models are getting more sophisticated, robust, and proficient in making precise predictions. However, they are becoming increasingly difficult to understand, resulting in a greater emphasis on the interpretability of ML, as it is essential to comprehend how these models arrive at their predictions. Understanding the rationale behind a decision or consistently predicting the outcome of a model is known as interpretability, according to [Miller, 2019]. The greater the interpretability of a machine learning model, the easier it becomes for humans to understand its outcomes. Interpretability aims to address inquiries such as: What variables influenced the prediction? To what extent does each feature impact the prediction? Illustrate the contribution of each feature to the prediction visually. This facilitates explaining the model's outcomes to peers, which is crucial from a scientific perspective. [Miller, 2019] proposes a logical framework comprising five elements: theory, model, types, entities, and data to enhance interpretability.

[Molnar, 2023] discusses the significance of interpretability in ML models. He argues that, in some cases, predicting an outcome alone does not solve the original problem one wants to understand. The author provides a diagram illustrating different levels of interpretability in ML models (refer to 6.1). A model is fully transparent at the highest level, and we can appreciate all the steps the model takes to make a prediction.

Figure 6.1: The big picture of explainable machine learning.


Word
Data
Black Box Model
Interpretability Methods
Humans
$Y=\alpha+\beta\cdot X$
IF x>4 THEN y=1
Composed of all variables, those that are observable are captured by the data.
Observable and cataloged variables or events are used by machine learning models for learning.
These models are used to predict variables of interest. Their results are extracted and interpreted using various methods.
They enable the interpretation of outcomes from machine learning models, helping humans understand the results.
Responsible for understanding and scientifically analyzing the results obtained.

Source: adapted from [Molnar, 2023]

For many reasons, understanding the interpretation of an ML model is crucial. Firstly, it can help us comprehend the connections among several variables. For instance, we may want to know if a particular variable is necessary for predicting another variable. Secondly, it can help us determine the factors that drive the model's predictions. For example, we might want to know if changes in the values of specific variables will impact the result. Thirdly, it can assist us in making better decisions. For instance, we might want to know what would happen if we had different observations. The more an ML model's decision affects an individual's life, the more critical it is for the model to explain its behavior. This is particularly true in economics, where ML models are frequently used to make decisions that significantly impact people's lives.

Despite the significance of interpretability, several challenges are associated with using ML in economics and finance. One of the challenges is that ML models can be complex, making it difficult to understand how they work and why they make predictions. Another challenge is that there is no single definition of interpretability, making it challenging to compare the interpretability of different ML models. Nonetheless, the use of ML in economics is expected to grow. As the ML models become more sophisticated, developing methods to make them more interpretable is becoming increasingly essential.

6.2 Model-Agnostic Artificial Intelligence Interpretation Methods

This section explores model-agnostic explainable artificial intelligence (XAI), also known as model-agnostic interpretation methods for machine learning models. The field of XAI focuses on developing techniques that enable humans to comprehend the predictions made by ML models or the decisions taken by Artificial intelligence systems. This field is not recent and has existed since artificial intelligence started. However, there has been an increased interest in XAI since the last decade due to companies and governments' increasing adoption of complex artificial intelligence solutions. Unlike model-specific methods and inherently interpretable models, model-agnostic approaches present a unique strategy. Model-specific methods are limited to certain types of models, and inherently interpretable models often trade off predictive accuracy for interpretability. As outlined by [Ribeiro et al., 2016], three essential qualities characterize artificial intelligence model-agnostic methods:

1. **Model Flexibility:** The interpretation method should be applicable to any machine learning model, spanning from random forests to deep neural networks.

2. **Explanation Flexibility:** The system should not be restricted to a singular form of explanation but rather accommodate various types, ranging from linear formulas to feature importance graphs.

3. **Representation Flexibility:** It is imperative that the explanation system can adapt to different feature representations than those utilized by the model under scrutiny. For instance, individual words can elucidate a text classifier operating on abstract word embeddings.

This versatility enables their wide-ranging applicability, from elementary decision trees to sophisticated deep learning architectures, and supports diverse forms of explanation, including linear formulas and feature importance graphs. Furthermore, their ability to adjust to different feature representations is essential.

XAI interpretability constitutes a crucial layer within the data processing hierarchy, commencing from real-world phenomena and traversing through data digitization, model construction, and interpretation. This approach merge the methodologies of statisticians, who emphasize data, and machine learning specialists, who contemplate the model layer, culminating in a comprehensive framework that enhances understanding and integrates the strengths of both disciplines.

Model-agnostic methods are categorized into local and global methods. Global methods analyze the average effects of features on predictions, while local methods elucidate individual predictions. This differentiation is pivotal for applying these techniques to diverse datasets, including intricate time series data. Commonly employed model-agnostic interpretability methods encompass:

- **Feature Importance Measures:** Identifying influential features without relying on the internal mechanics of the model. Examples include permutation feature importance and SHAP values.
- **Partial Dependence Plots (PDP):** Visualizing the impact of specific features on model predictions.
- **Local Interpretable Model-agnostic Explanations (LIME):** Explicating individual predictions by approximating the model with a simpler one.
- **Global Surrogate Models:** Utilizing simpler models such as decision trees to interpret complex model predictions.
- **Counterfactual Explanations:** Understanding model behavior through hypothetical scenarios.
- **Individual Conditional Expectation (ICE) Plots:** Providing detailed insights into how changes in features influence individual predictions.
- **Sensitivity Analysis:** Assessing model robustness by observing prediction variations with slight input perturbations.
- **Anchors:** Identifying critical features that significantly impact specific predictions.

These versatile tools can be effectively deployed across a spectrum of models, from basic regressions to intricate neural networks, rendering them indispensable in the realm of machine learning interpretation.

6.2.1 Understanding Shapley Values

A pivotal technique within the domain of the model-agnostic explainable artificial intelligence method is the Shapley value. According to [Molnar, 2023], these methods are esteemed for adaptability, as they can be applied universally across diverse ML models. Consequently, they facilitate the development of derivative works, such as graphical interfaces, liberated from model dependencies.

The Shapley Value concept is based on game theory. In a cooperative game framework, each predictor variable is like a "player" that contributes to the target variable's prediction or "payout." The concept was developed by [Shapley, 2016] to show when cooperation among players results in a higher overall payoff compared to individual actions, and how the surplus payoff is divided among the players who have different weights or relevance in the coalition. In the context of machine learning and explainable AI, the explanatory variables or features act like players that contribute differently to the model's prediction. The decision function plays the role of the coalition.

As articulated by [Molnar, 2023], the underlying principle of the Shapley value is relatively intuitive: a predictor variable "enters" a room at random and collaborates in a game to contribute to the overall prediction. The Shapley value of a given predictor is essentially the average incremental change it brings to the projections made by the current coalition of predictors.

Mathematically, the Shapley Value φ_j for a predictor j is computed as follows:

$$\varphi_j(\mathrm{val}) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\mathrm{val}(S \cup \{j\}) - \mathrm{val}(S)\right) \tag{6.1}$$

Here, S is a subset of the predictors considered in the model, x represents the vector of predictor values under scrutiny, and p is the total number of predictors. The function $\mathrm{val}_x(S)$ calculates the prediction for the predictors included in S while marginalizing over the excluded predictors:

$$\mathrm{val}_X(S) = \int \widehat{f}(x_1, \ldots, x_p) d\mathbb{P}_X - E_X(\widehat{f}(X)) \tag{6.2}$$

Evaluating the Shapley Value necessitates multiple integrations for each predictor, thereby establishing its cooperative effect on the forecast. The Shapley value adheres to four fundamental properties: efficiency, symmetry, dummy, and additivity.

- **Efficiency**: The sum of all Shapley values is equal to the difference between the forecast for x and the expected forecast:

$$\sum_{j=1}^{p} \varphi_j = \widehat{f}(x) - E_X(\widehat{f}(X)) \tag{6.3}$$

- **Symmetry**: Identical contributions from predictors j and k yield the same Shapley value:

$$\mathrm{val}(S \cup \{j\}) = \mathrm{val}(S \cup \{k\}) \Rightarrow \varphi_j = \varphi_k \tag{6.4}$$

- **Dummy**: A predictor with no impact has a Shapley value of zero:

$$\mathrm{val}(S \cup \{j\}) = \mathrm{val}(S) \Rightarrow \varphi_j = 0 \tag{6.5}$$

- **Additivity**: For games with a total payout of $\mathrm{val} + \mathrm{val}^{+}$, the Shapley values can be additive:

$$\varphi_j + \varphi_j^+ \tag{6.6}$$

For ensembles like decision tree forests, determining individual contributions is computationally challenging. As a workaround, Monte Carlo approximations offer estimated Shapley values:

$$\widehat{\varphi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\widehat{f}(x_{+j}^m) - \widehat{f}(x_{-j}^m) \right) \tag{6.7}$$

In this formula, $\widehat{f}(x_{+j}^m)$ signifies the forecast with predictor j while randomizing other predictors, and M symbolizes the number of Monte Carlo samples.

This "Frankenstein's Monster" approach, as coined by Molnar [Molnar, 2023], amalgamates different random samples to approximate the Shapley value. While the Shapley value offers fair value distribution among predictors and can elucidate individual or group contributions, its major drawback lies in its computational intensity, often necessitating approximation methods.

6.2.2 Opening the "Black Box" with Shapley Values

In Chapter 5, we discussed the predictions made by our machine learning models. After analyzing the results, we found that the Gradient Boosting model had the best accuracy metrics and passed the MCS hypothesis test. The model effectively predicted our target, but we still have questions about the rationale behind its predictions. Specifically, we want to know how much each feature contributed to the variance between the final prediction and the average prediction. In this sense, to better understand the outcomes derived from this model, we employed the Shapley Additive Explanation (SHAP) approach. This technique, introduced by [Lundberg & Lee, 2017], uses cooperative game theory to explain individual model predictions. Its model-agnostic nature sets the SHAP approach apart from other interpretation methods. It is not specific to any particular model and integrates different additive feature attribution methods, such as Local Interpretable Model-Agnostic Explanations (LIME).

We used the *SHAP* (Shapley Additive exPlanations) package, available in *Python*, to analyze the Gradient Boosting results. Although computationally intensive, this package provides a detailed analysis of the variable effects on outcome predictions. We can better understand their impact on the training set by dissecting and analyzing each variable's role. It is good practice to look at the training set results using the Shapley Value approach, as they provide more information about the model's performance than predicted values.

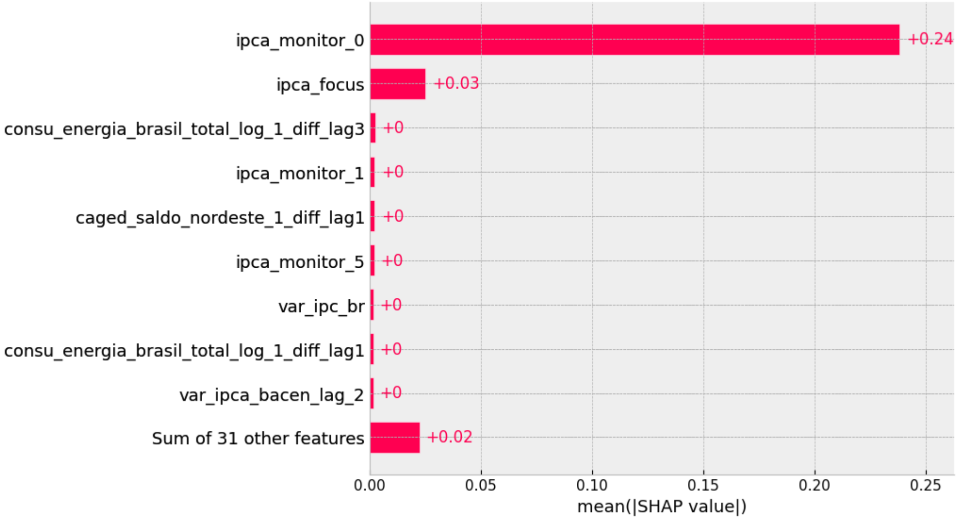


Figure 6.2: SHAP average absolute impact on output

We begin our analysis by exploring the global significance of features using the SHAP (SHapley Additive exPlanations) package. Figure 6.2 illustrates this analysis. As outlined by [Molnar, 2023], the computation of SHAP feature importance involves averaging the absolute Shapley values for each feature. This approach allows us to ascertain the global importance of features by calculating the mean of the absolute Shapley values across the dataset. It is crucial to distinguish that SHAP feature importance fundamentally differs from permutation feature importance. The latter assesses feature significance based on the decrease in model performance, whereas SHAP focuses on the magnitude of feature contributions. The chart in Figure 6.2 presents the average absolute Shapley values for the key variables.

We can observe differences among the features selected via the Shapley Value approach and the ElasticNet method. The Shapley Value approach identifies the variable ‘ipca_monitor_0‘ as the most significant, contributing an average of 0.24 absolute points to the overall model prediction. The ElasticNet method corroborates this finding, also recognizing this variable as having the highest coefficient and appearing most frequently in our RFE approach. However, disparities emerge when examining the second most crucial feature onward. According to the Shapley Value method, ‘ipca_focus‘ is the second most relevant, with an average contribution of 0.03. In contrast, ElasticNet ranks ‘var_ipc_fipe‘ as the second most significant based on its coefficient level. Moreover, ElasticNet indicates that several other features hold substantial average coefficients, thus contributing more significantly to model performance than indicated by the Shapley Value method. These differences are detailed in Chart 5.7 and the table in Appendix B, which list the ElasticNet coefficients for each selected variable. Additionally, when evaluating the frequency criterion in ElasticNet through the RFE approach, ‘ipca_focus‘ appears as

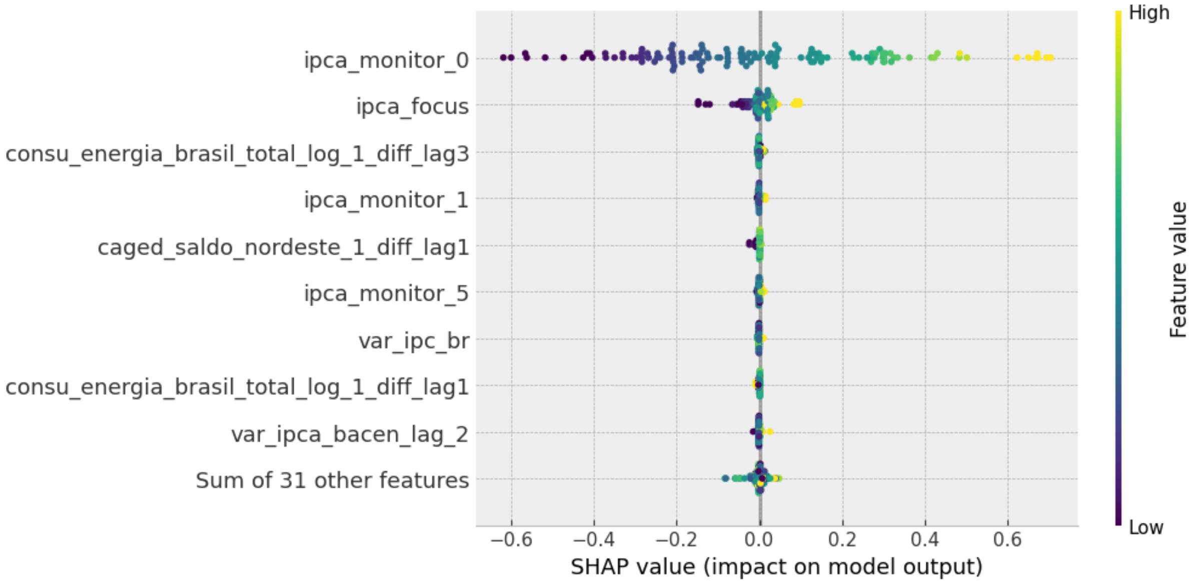


Figure 6.3: Average impact on model output magnitude

the second most frequent feature, as detailed in the table in Appendix C.

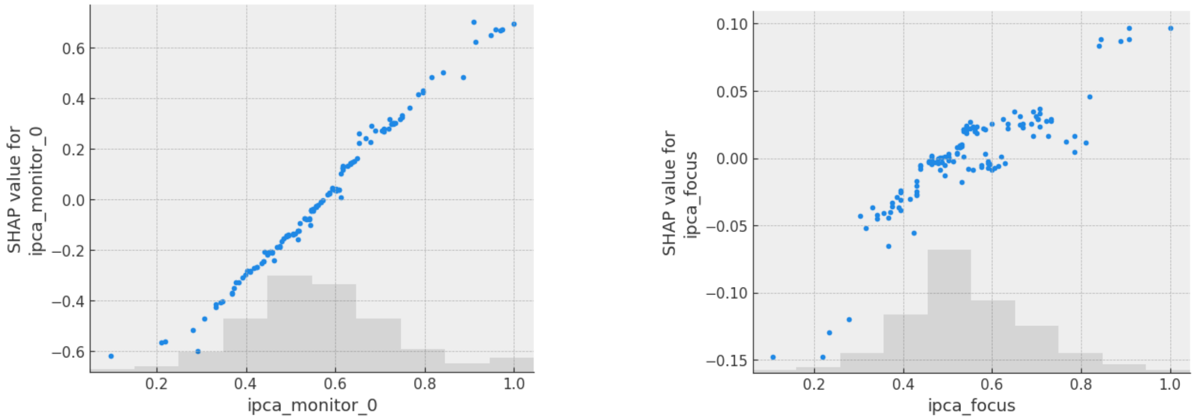
To summarize, ElasticNet's feature selection indicates that the second most important feature based on coefficient value is ‘var_ipc_fipe,‘ with ‘ipca_focus‘ being the second most frequently selected feature. Conversely, the Shapley Value method assigns ‘ipca_focus‘ the role of the second most crucial contributor to model outcomes. It's worth noting that 'ipca_monitor_0' and 'ipca_focus' play a vital role in both methods.The reason why these features are significant is due to their high correlation with our target. 'ipca_monitor_0' aims to replicate the target, while 'ipca_focus' represents the median market expectation of the Focus-Market Readout participants released by the BCB every week. In other words, they both try to capture the target dynamics.

Continuing our exploration of global importance, we can visualize the SHAP values assigned to each feature across all samples, offering insights into the most crucial features' role in prediction. Transforming local importance into a global perspective, we leverage tools like Beeswarm plot as a compelling alternative for extracting overarching insights. Chart 6.3 offers a detailed summary of how the key features in a dataset influence the model's predictions, merging the concepts of feature importance and their effects. In this plot, each dot represents the Shapley value of a feature for a specific instance, where the feature's role is denoted on the y-axis and its Shapley value on the x-axis. The dots' color gradient from low (purple) to high (yellow) values illustrates the feature's impact.

The overlap points are slightly spread out vertically to reveal the distribution of Shapley values for each feature, arranged by their significance. This chart reinforces that 'ipca_monitor_0' has the most crucial role, in line with the chart 6.2 and a significant spread related to the impact on the output with its Shapely Value ranging from -0.7 to +0.7. The second most important is the 'ipca_focus' have a minor effect with Shapley values ranging from -0.2 to 0.2, and also, there is a direct relation between feature importance and the Shapley value. The other variables have Shapely values next to zero, meaning they have little impact on the output.

The Beeswarm plot provides insights into the connection between the feature value and its effect on each prediction point. However, we need to examine the SHAP dependence plots to understand the relationship precisely for a singular feature and its SHAP value. A dependence plot is a visualization tool that displays the relationship between a feature's SHAP value and its corresponding feature value. This plot contains a set of points where each point represents an observation in the data set. The horizontal axis of the chart shows the actual value of the feature, while the vertical axis represents the contribution of the feature to the SHAP value. The dependence plot is instrumental if the feature has a non-linear relationship with the target variable. As illustrated in Figure 6.4 for 'ipca_monitor_0' (a) and 'ipca_focus' (b), respectively, the dependence plot shows a direct relation between the variable values with its SHAP values. This relation is not as evident for the second variable as the first; there is a linear dependence for 'ipca_focus' but with some concentration between -0.5 e +0.5 for its SHAP value and between 0.4 and 0.8 for its values.

Figure 6.4: Individual dependence plot to ipca_monitor_0 and ipca_focus



a) Dependence plot of ipca_monitor_0

b) Dependence plot of ipca_focus

Another way to analyze the dependence plot is by looking at the interaction effect, where we plot together the most crucial features and see how the values of one affect the other's Shapely Value. According to [Molnar, 2023], the Shapley interaction index from game theory is defined as:

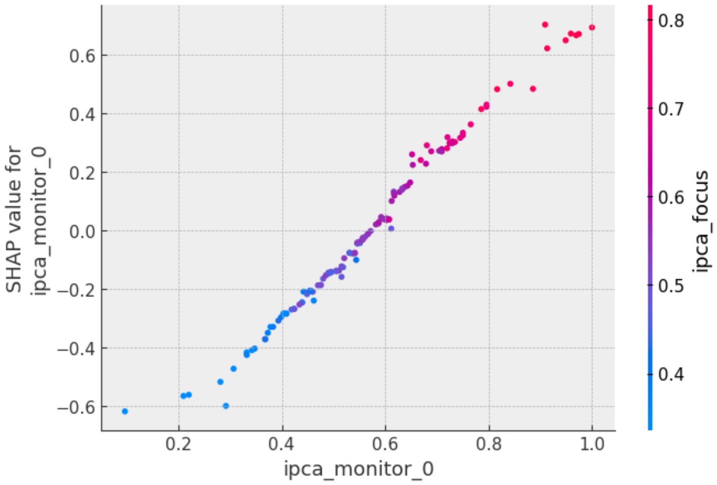


Figure 6.5: Interaction effect

$$\phi_{i,j} = \sum_{S \subseteq \setminus \{i,j\}} \frac{|S|!(M - |S| - 2)!}{2(M-1)!} \delta_{ij}(S)$$

when $i \neq j$ and

$$\delta_{ij}(S) = \hat{f}_x(S \cup \{i,j\}) - \hat{f}_x(S \cup \{i\}) - \hat{f}_x(S \cup \{j\}) + \hat{f}_x(S)$$

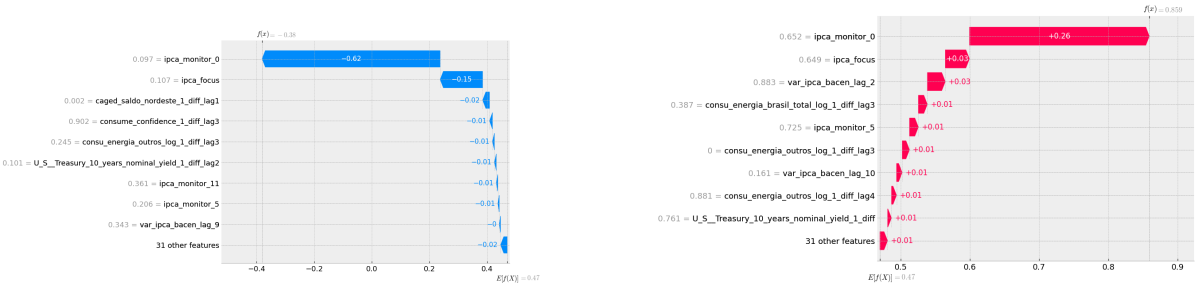
As outlined in Molnar's work [Molnar, 2023], this formula elucidates the interaction effect of features by accounting for their main effects and integrating individual effects.

Figure 6.5 demonstrates the interaction effect between our analysis's two most significant features. The structure is the same as the previous dependent plot, but the color scale in right side reflects the values of an interaction variable which in our case is the 'ipca_focus'. The color gradient clearly illustrates a direct relationship between the two features. As 'ipca_monitor_0' values increase, the corresponding 'ipca_focus' values also tend to rise (tend to have a red color), and similarly, lower values of the first imply lower values of the second. Notably, both variables have negative SHAP values when their values are close to zero. However, this relationship shifts to positive when the values of 'ipca_monitor_0' exceed 0.6 and 'ipca_focus' surpasses 0.5.

Finally, in Figure 6.6, we present local explainability plots for two observations, one with a negative prediction on the left chart and another with a positive prediction on the right chart. Each chart displays the predicted value after considering all features, denoted by $f(x)$. For the left chart, this value is -0.380, and for the right chart, it is 0.859. The mean prediction, indicated by $E[f(x)]$, remains constant at 0.47 in both cases.

Within the plots in figure 6.6, the blue bars represent how much a specific feature reduces the predictive value, while the red bars signify the extent to which a particular feature increases it. The total contribution is the difference between the prediction ($f(x)$) and the mean prediction ($E[f(x)]$). As anticipated, our two most critical variables, 'ipca_monitor_0' and 'ipca_focus,' exerted the most significant influence on both negative and positive values. In the negative prediction, 'ipca_monitor_0' decreased the prediction by 62 basis points, while 'ipca_focus' reduced it by 15 basis points from the average value of 0.47, as illustrated in the chart. Conversely, for the positive prediction, 'ipca_monitor_0' contributed 26 basis points, while 'ipca_focus' increased the prediction by three basis points.

Figure 6.6: Local plot to negative and positive prediction



a) Local plot to a negative prediction

b) Local plot to a positive prediction

Following the two most influential features, there is a notable variability among other features reported in the two local plots. While individually these features may have only a marginal impact on predictions, their collective contribution proves significant to both sets of results. It is imperative to underscore the role of lagged variables in influencing outcomes; notably, 'var_ipca_bacen' emerges among the top ten most crucial features with varying lag intervals in both plots. In chart 6.6.a, it appears with nine lags, whereas in chart 6.6.b, it manifests with two lags. This underscores the imperative for feature engineering when employing machine learning techniques to enhance performance.

Based on the insights revealed by SHAP values, it is evident that proxy variables such as 'ipca_monitor_0' and 'ipca_focus' play a pivotal role in achieving optimal outcomes when utilizing ML models. While features associated with economic activity, the labor market, and the financial market do contribute to the model's performance, their impact is not as pronounced in yielding superior results. These findings hold significant implications, particularly when leveraging predictive results to formulate strategies for a Treasury desk within a major financial institution. In such contexts, the ability to attain the most effective outcomes is crucial.

Chapter 7

Conclusions

This study aimed to identify the most effective nonlinear machine learning model for predicting the monthly IPCA. We employed a feature selection method based on the shrinkage technique to isolate the most critical predictors. Additionally, our research utilized a model-agnostic explainable artificial intelligence (XAI) method known as Shapley Value to provide us with valuable insights into ML model predictions, often regarded as "black boxes."

Our journey began collecting data from IPCA between August 2010 and January 2024 and creating a dataset consisting of 156 variables as predictors based on the literature review, ensuring a rich and varied foundation for analysis. This comprehensive dataset was the foundation for our predictions of the Brazilian consumer price index (IPCA) from May 2021 to January 2024. Previously, to predict, we conducted a target analysis to determine our study's most effective modeling approach. This step directed our application of ML techniques, particularly the linear ElasticNet method. Utilizing this shrinkage method, we refined our dataset by applying the Recursive Feature Elimination (RFE) method to select the 30 most significant variables to use as predictors.

The core of our study was applying these refined features to four nonlinear forecasting models, including Random Forest, AdaBoost, Extreme Gradient Boosting (XGBoost), and Gradient Boosting. These models were specifically chosen for their proficiency in capturing complex, nonlinear interactions often overlooked by linear models. Utilizing a comprehensive evaluation by multiple accuracy metrics, residual analysis, and hypothesis testing through the Model Confidence Set (MCS) method, Gradient Boosting emerged as the most effective model, showcasing superior performance in terms of minimal MAE, MSE and RMSE.

The interpretability of Gradient Boosting was enhanced by applying Shapley Value, an emerging method of Explainable Artificial Intelligence (XAI). This method sheds light on the contribution of each feature to the prediction outcomes, addressing concerns about the interpretability of ML models. It provides valuable insights into the dynamics of feature interactions. Using XAI methods helps reflect on theory and its empirical applications for forecasting. Understanding the reasoning behind specific model results enhances the usefulness of research findings that use ML models.

During our literature review, we came across various ML algorithms that have the potential to predict inflation. However, due to the time constraints of a master's degree, we could only explore a limited number of models. For future works using data from the Brazilian economy, there is an opportunity to explore other models that may perform better in forecasting our target. Deep learning or Hybrid models could be promising for predictive analytics and warrant exploration within inflation forecasting or to solve macroeconometric problems. Integrating machine learning methodologies with traditional macroeconomic tools can be highly beneficial. Additionally, XAI methods, particularly the Shapley Value approach, provide a valuable resource for future applications in macroeconometric literature and financial markets where understanding inflation prediction is crucial for effective market risk management.

Utilizing insights from SHAP values and Feature Selection, this study highlights the crucial role of proxy of our target like 'ipca_monitor_0' and 'ipca_focus' in improving results with ML models. While variables related to economic activity, the labor market, and financial markets influence model performance, their impact is not as significant as that of the proxy variables. This has important implications for strategy development within a financial institution's Treasury desk. Recognizing the importance of these proxy variables is key to achieving optimal outcomes, and the methods used in this work showed it.

In summary, our research significantly contributes to economic forecasting by showcasing the efficacy of nonlinear models in predicting inflation in an emerging country. According to [Blanchard, 2018], learning algorithms are considered good forecasting models. For the author, reference in macroeconomics, a good forecasting model is the one that has the most accurate forecast rather than the one that is more in line with macroeconomic theory. It means that if the theory is irrelevant to forecasting, it should be discarded. Accordingly, our findings illuminate a promising avenue for refining inflation forecasts by incorporating sophisticated machine-learning techniques that are more reliable in an environment where big data is becoming more and more common in the finance and economic fields. These include advanced feature selection and the enhancement of model interpretability through innovative XAI methods. Such advancements pave the way for developing more nuanced and precise forecasting methodologies. Moreover, beyond contributing to academic debate, our research provides practical insights for policymakers and financial institutions, equipping them with reliable economic forecasts to navigate the complexities of a volatile economic environment.

References

[Acosta, 2018] Acosta, M. (2018). Machine learning core inflation. *ECONOMICS LETTERS*, 169, 47–50. 11, 12

[Alfiyatin et al., 2019] Alfiyatin, A., Rizki, A., Mahmudy, W., & Ananda, C. (2019). Extreme learning machine and particle swarm optimization for inflation forecasting. *International Journal of Advanced Computer Science and Applications*, 10(4), 473–478. 1 citations (Crossref) [2023-08-08]. 10, 12

[Amat Rodrigo & Escobar Ortiz, 2023] Amat Rodrigo, J. & Escobar Ortiz, J. (2023). *skforecast*. skforecast. 56

[Aras, 2022] Aras, L. (2022). Explainable inflation forecasts by machine learning models. 3, 12, 16

[Araujo & Gaglianone, 2023] Araujo, G. & Gaglianone, W. (2023). Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. *Latin American Journal of Central Banking*, 4(2). 11, 12, 44, 52

[Athey & Imbens, 2019] Athey, S. & Imbens, G. (2019). Machine Learning Methods That Economists Should Know About. *Annual Review of Economics*, 11(1), 685–725. Publisher: Annual Reviews. 9, 25, 31

[Behrens et al., 2018] Behrens, C., Pierdzioch, C., & Risse, M. (2018). Testing the optimality of inflation forecasts under flexible loss with random forests. *ECONOMIC MODELLING*, 72, 270–277. 22 citations (Crossref) [2023-08-08]. 10, 12

[Bergmeir et al., 2018] Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120, 70–83. 31

[Bergstra & Bengio, 2012] Bergstra, J. & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10), 281–305. 34

[Blanchard, 2018] Blanchard, O. (2018). On the future of macroeconomic models. *Oxford Review of Economic Policy*, 34(1-2), 43–54. 76

[Breiman, 2001] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. 25

[Buckmann et al., 2021] Buckmann, M., Joseph, A., & Robertson, H. (2021). Opening the Black Box: Machine Learning Interpretability and Inference Tools with an Application to Economic Forecasting. In S. Consoli, D. Reforgiato Recupero, & M. Saisana (Eds.), *Data Science for Economics and Finance: Methodologies and Applications* (pp. 43–63). Cham: Springer International Publishing. 12, 15

[Cakan et al., 2015] Cakan, E., Doytch, N., & Upadhyaya, K. P. (2015). Does U.S. macroeconomic news make emerging financial markets riskier? *Borsa Istanbul Review*, 15(1), 37–43. 1

[CFA, 2023] CFA, I. (2023). Measuring and Managing Market Risk. 1

[Chen & Guestrin, 2016] Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754. 27

[Coulombe et al., 2022] Coulombe, P., Leroux, M., Stevanovic, D., & Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *JOURNAL OF APPLIED ECONOMETRICS*, 37(5), 920–964. 12, 15

[Diebold & Mariano, 1995] Diebold, F. & Mariano, R. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–63. Publisher: American Statistical Association. 61

[Fisher, Irving, 1930] Fisher, Irving (1930). The Theory of Interest: As Determined by Impatience to Spend Income and Opportunity to Invest It. Accession Number: 6255 Source: oclc. 2

[Fonti & Belitser, 2017] Fonti, V. & Belitser, E. (2017). Feature selection using lasso. *VU Amsterdam research paper in business analytics*, 30, 1–25. 12

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. Publisher: Institute of Mathematical Statistics. 26

[Friedman & Hall, 2007] Friedman, J. H. & Hall, P. (2007). On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference*, 3(137), 669–683. 25

[Garcia et al., 2017] Garcia, M. G. P., Medeiros, M., & Vasconcelos, G. F. R. (2017). Real-time inflation forecasting with high-dimensional models: The case of Brazil. *International Journal of Forecasting*, 33(3), 679–693. Publisher: Elsevier. 2, 11, 12, 44

[García-Portugués, 2024] García-Portugués, E. (2024). *4.1 Shrinkage | Notes for Predictive Modeling*. BookDown. 22

[Geron, 2019] Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd edition. 18, 36, 48

[Granger & Newbold, 1974] Granger, C. W. J. & Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics*, 2(2), 111–120. 47

[Hansen et al., 2011] Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The Model Confidence Set. *Econometrica*, 79(2), 453–497. Publisher: [Wiley, Econometric Society]. 36, 61

[Hyndman & Athanasopoulos, 2021] Hyndman, R. J. & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice (2nd ed)*. Otexts. 40

[imf, 2023] imf, g. (2023). Global Financial Stability Report, April 2023. 4

[Iskhakov et al., 2020] Iskhakov, F., Rust, J., & Schjerning, B. (2020). Machine learning and structural econometrics: contrasts and synergies. *The Econometrics Journal*, 23(3), S81–S124. 3, 9

[James et al., 2013] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer. 25

[James et al., 2023] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in python*. Springer Nature. 20

[Joseph et al., 2021] Joseph, A., Kalamara, E., Kapetanios, G., & Potjagailo, G. (2021). Forecasting UK inflation bottom up. *International Journal of Forecasting*. 1 citations (Crossref) [2023-08-08]. 12, 15

[Kim & Swanson, 2018] Kim, H. & Swanson, N. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting*, 34(2), 339–354. 62 citations (Crossref) [2023-08-08]. 14

[Kleiber & Zeileis, 2008] Kleiber, C. & Zeileis, A. (2008). *Applied Econometrics with R*. New York, NY: Springer New York. 42

[Lazzeri, 2020] Lazzeri, F. (2020). *Machine Learning for Time Series Forecasting with Python®*. Wiley, 1 edition. 36, 39, 56

[Li, 2024] Li, H. L. a. M. (2024). *10.3 Variable selection property of the lasso | Introduction to Data Science*. Data Science Workshop. 20, 21

[Lu & Wu, 2009] Lu, B. & Wu, L. (2009). Macroeconomic releases and the interest rate term structure. *Journal of Monetary Economics*, 56(6), 872–884. 2

[Lundberg & Lee, 2017] Lundberg, S. & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs, stat]. 69

[Mariano & Santos, 2017] Mariano, A. & Santos, M. (2017). (PDF) Revisão da Literatura: Apresentação de uma Abordagem Integradora. 6, 7

[Masini et al., 2021] Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2021). Machine Learning Advances for Time Series Forecasting. arXiv:2012.12802 [cs, econ, stat]. 9

[Medeiros et al., 2021] Medeiros, M. C., Vasconcelos, G. F., Veiga, , & Zilberman, E. (2021). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, 39(1), 98–119. Publisher: Taylor & Francis. 10, 12, 31

[Miller, 2019] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1–38. 64

[Molnar, 2023] Molnar, C. (2023). Interpretable Machine Learning. 64, 65, 67, 68, 69, 70, 72, 73

[nielsaka, 2023] nielsaka (2023). nielsaka/modelconf. original-date: 2018-08-03T12:26:29Z. 63

[Nosratabadi et al., 2020] Nosratabadi, S., Mosavi, A., Duan, P., Ghamisi, P., Filip, F., Band, S. S., Reuter, U., Gama, J., & Gandomi, A. H. (2020). Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods. *Mathematics*, 8(10), 1799. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute. 10, 31

[Ouyang et al., 2021] Ouyang, Z., Ravier, P., & Jabloun, M. (2021). STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not by Machine Learning Ones. *Engineering Proceedings*, 5(1), 42. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute. 41

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. 48, 56

[Peng & Nagata, 2020] Peng, Y. & Nagata, M. H. (2020). An empirical overview of nonlinearity and overfitting in machine learning using COVID-19 data. *Chaos, Solitons & Fractals*, 139, 110055. Publisher: Elsevier. 3

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-Agnostic Interpretability of Machine Learning. arXiv:1606.05386 [cs, stat]. 66

[Riofrío et al., 2020] Riofrío, J., Chang, O., Revelo-Fuelagán, E., & Peluffo-Ordóñez, D. (2020). Forecasting the Consumer Price Index (CPI) of Ecuador: A comparative study of predictive models. *International Journal on Advanced Science, Engineering and Information Technology*, 10(3), 1078–1084. 10, 12

[Shapley, 2016] Shapley, L. S. (2016). 17. A Value for n-Person Games. In *17. A Value for n-Person Games* (pp. 307–318). Princeton University Press. 67

[Skforecast, 2023] Skforecast (2023). Introduction to forecasting - Skforecast Docs. 36, 37

[Skforecast, 2024] Skforecast (2024). Tuning forecaster - Skforecast Docs. 56

[Sklearn, 2024] Sklearn (2024). L1-based models for Sparse Signals. 22

[Spiess & Neumeyer, 2010] Spiess, A.-N. & Neumeyer, N. (2010). An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. *BMC Pharmacology*, 10(1), 6. 60

[Stańczyk & Jain, 2015] Stańczyk, U. & Jain, L. C., Eds. (2015). *Feature Selection for Data and Pattern Recognition*, volume 584 of *Studies in Computational Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg. 13, 14

[Syed & Lee, 2021] Syed, A. & Lee, K. (2021). Macroeconomic forecasting for Pakistan in a data-rich environment. *Applied Economics*, 53(9), 1077–1091. 1 citations (Crossref) [2023-08-08]. 11, 12

[Tibshirani, 1996] Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. 19

[Tibshirani, 2014] Tibshirani, R. J. (2014). Lasso and sparsity in statistics. *Academia*. 19

[Tibshirani & Friedman, 2001] Tibshirani, S. & Friedman, H. (2001). The Elements of Statistical Learning. *Academia*. 29

[Tibshirani & Wainwright, 2016] Tibshirani, S. & Wainwright, J. (2016). Valerie and Patrick Hastie. *Academia*. 17

[Verleysen & François, 2005] Verleysen, M. & François, D. (2005). *The Curse of Dimensionality in Data Mining and Time Series Prediction*, volume 3512. Computational Intelligence and Bioinspired Systems. 3

[Wang et al., 2023] Wang, C. S. H., Fan, R., & Xie, Y. (2023). Market systemic risk, predictability and macroeconomics news. *Finance Research Letters*, 56, 104102. 1

[Wang et al., 2006] Wang, X., Smith-Miles, K., & Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.*, 13, 335–364. 40

[Wang et al., 2020] Wang, Z., Zhu, Z., & Yu, C. (2020). Variable Selection in Macroeconomic Forecasting with Many Predictors. arXiv:2007.10160 [econ, stat]. 22, 36

[Yilmazkuday, 2022] Yilmazkuday, H. (2022). Inflation and growth: the role of institutions. *Journal of Economics and Finance*, 46(1), 167–187. 4

[Zach, 2020] Zach (2020). How to Perform a Mann-Kendall Trend Test in R. 40

[Zaman et al., 2023] Zaman, M., Tania, N. N., & Khan, M. S. A. (2023). Treasury Management in Financial Institutions. *The Institute of Bankers, Bangladesh*. 1

[Zhao & Yu, 2006] Zhao, P. & Yu, B. (2006). On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*. 22, 23

[Zou & Hastie, 2005] Zou, H. & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2), 301–320. Publisher: [Royal Statistical Society, Wiley]. 23

[Özgür & Akkoç, 2021] Özgür, & Akkoç, U. (2021). Inflation forecasting in an emerging economy: selecting variables with machine learning algorithms. *International Journal of Emerging Markets*, 17(8), 1889–1908. Publisher: Emerald Publishing Limited. 12, 14, 22, 36, 49, 52

Appendix A

Stationarity Tests

Table 10: Stationarity Tests Results (P Value)

ADF	PP	KPSS	Stationarity Decision	Series
0,01	0,01	0,09	accepted	var_ipc_fipe
0,01	0,01	0,13	accepted	var_ipca_15
0,01	0,01	0,20	accepted	var_ipc_br
0,11	0,01	0,58	rejected	var_ipc_br_core
0,10	0,01	0,13	rejected	var_igp_m
0,06	0,01	0,13	rejected	var_igp_di
0,06	0,01	0,14	rejected	var_igp_10
0,01	0,01	0,13	accepted	ipca_monitor_1
0,01	0,01	0,10	accepted	ipca_monitor_11
0,07	0,01	1,48	rejected	ipca_monitor_12
0,01	0,01	0,15	accepted	ipca_monitor_2
0,20	0,01	0,19	rejected	ipca_monitor_3
0,50	0,01	0,40	rejected	ipca_monitor_4
0,01	0,01	0,06	accepted	ipca_monitor_5
0,01	0,01	0,29	accepted	ipca_monitor_6
0,01	0,01	2,45	rejected	ipca_monitor_7
0,01	0,01	0,16	accepted	ipca_monitor_8
0,01	0,01	0,13	accepted	ipca_monitor_9
0,01	0,01	0,23	accepted	ipca_monitor_0
0,01	0,01	0,17	accepted	ipca_focus
0,17	0,01	0,40	rejected	VIX
0,30	0,01	0,44	rejected	VIX_log
0,01	0,01	0,05	accepted	MSCI_emerging_1_diff
0,01	0,01	0,04	accepted	MSCI_emerging_log_1_diff
0,01	0,01	0,09	accepted	NOAA_Index_1_diff
0,23	0,01	0,38	rejected	U.S..Treasury.3.months.nominal.yield_1_diff
0,05	0,01	0,21	rejected	U.S..Treasury.2.years.nominal.yield_1_diff

0,01	0,01	0,18	accepted	U.S..Treasury.10.years.nominal.yield_1_diff
0,01	0,01	0,08	accepted	BRL_USD_1_diff
0,01	0,01	0,12	accepted	BRL_USD_log_1_diff
0,01	0,01	0,03	accepted	VIX_1_diff
0,01	0,01	0,04	accepted	VIX_log_1_diff
0,01	0,01	0,05	accepted	DXY_1_diff
0,01	0,01	0,06	accepted	DXY_log_1_diff
0,01	0,01	0,17	accepted	Ibovespa_1_diff
0,01	0,01	0,16	accepted	Ibovespa_log_1_diff
0,01	0,01	0,07	accepted	Brasil_CDS5Y_1_diff
0,01	0,01	0,08	accepted	Brasil_CDS5Y_log_1_diff
0,01	0,01	0,09	accepted	crb_1_diff
0,01	0,01	0,10	accepted	crb_food_1_diff
0,01	0,01	0,06	accepted	crb_metals_1_diff
0,01	0,01	0,09	accepted	crb_log_1_diff
0,01	0,01	0,11	accepted	crb_food_log_1_diff
0,01	0,01	0,06	accepted	crb_metals_log_1_diff
0,01	0,01	0,08	accepted	oil_price_brent_1_diff
0,01	0,01	0,07	accepted	oil_price_wti_1_diff
0,01	0,01	0,07	accepted	oil_price_brent_log_1_diff
0,01	0,01	0,06	accepted	oil_price_wti_log_1_diff
0,31	0,01	0,18	rejected	selic_aa_1_diff
0,31	0,01	0,18	rejected	selic_aa_2_diff
0,01	0,01	0,12	accepted	var_ipca_bacen_lag1
0,04	0,01	0,14	accepted	var_comercializaveis_bacen_lag1
0,01	0,01	0,43	rejected	var_nao_comercializaveis_bacen_lag1
0,01	0,01	0,07	accepted	var_administrados_bacen_lag1
0,01	0,01	0,10	accepted	var_bens_nao_duraveis_bacen_lag1
0,66	0,01	0,30	rejected	var_bens_semi_duraveis_bacen_lag1
0,31	0,01	0,58	rejected	var_duraveis_bacen_lag1
0,01	0,01	0,96	rejected	var_servicos_bacen_lag1
0,19	0,01	0,26	rejected	var_medias_aparadas_sem_suavizacao_bacen_lag1
0,02	0,01	0,35	accepted	var_nucleo_por_exclusao_ex0_bacen_lag1
0,01	0,01	0,21	accepted	var_livres_bacen_lag1
0,04	0,01	0,27	accepted	var_nucleo_por_exclusao_ex1_bacen_lag1
0,15	0,01	0,30	rejected	var_nucleo_dupla_ponderacao_bacen_lag1
0,06	0,01	0,31	rejected	var_difusao_bacen_lag1
0,43	0,01	0,45	rejected	var_nucleo_por_exclusao_ex2_bacen_lag1
0,41	0,01	0,47	rejected	var_nucleo_por_exclusao_ex3_bacen_lag1
0,13	0,01	0,20	rejected	var_industriais_bacen_lag1
0,01	0,01	0,08	accepted	var_alimentacao_em_domicilio_bacen_lag1
0,31	0,01	0,38	rejected	var_nucleo_percentil_55_bacen_lag1
0,03	0,01	0,28	accepted	var_nucleo_ex_alimentacao_energia_exfe_bacen_lag1

0,01	0,01	0,11	accepted	var_incc_lag1
0,06	0,01	2,50	rejected	consu_energia_brasil_total_lag1
0,01	0,01	2,93	rejected	consu_energia_residencia_lag1
0,08	0,01	2,88	rejected	consu_energia_outros_lag1
0,01	0,01	0,67	rejected	caged_saldo_norte_lag1
0,01	0,01	0,41	rejected	caged_saldo_nordeste_lag1
0,03	0,01	0,42	rejected	caged_saldo_sudeste_lag1
0,04	0,01	0,30	accepted	caged_saldo_sul_lag1
0,01	0,01	0,52	rejected	caged_saldo_centro_oeste_lag1
0,04	0,01	0,48	rejected	caged_saldo_brasil_lag1
0,04	0,01	2,49	rejected	consu_energia_brasil_total_log_lag1
0,01	0,01	2,93	rejected	consu_energia_residencia_log_lag1
0,11	0,01	2,83	rejected	consu_energia_outros_log_lag1
0,01	0,01	0,04	accepted	MSCI_emerging_1_diff_lag1
0,01	0,01	0,04	accepted	MSCI_emerging_log_1_diff_lag1
0,01	0,01	0,09	accepted	NOAA_Index_1_diff_lag1
0,23	0,01	0,39	rejected	U.S..Treasury.3.months.nominal.yield_1_diff_lag1
0,06	0,01	0,22	rejected	U.S..Treasury.2.years.nominal.yield_1_diff_lag1
0,01	0,01	0,18	accepted	U.S..Treasury.10.years.nominal.yield_1_diff_lag1
0,01	0,01	0,30	accepted	PNAD_DESOC_1_diff_lag1
0,01	0,01	0,08	accepted	BRL_USD_1_diff_lag1
0,01	0,01	0,11	accepted	BRL_USD_log_1_diff_lag1
0,01	0,01	0,04	accepted	VIX_1_diff_lag1
0,01	0,01	0,05	accepted	VIX_log_1_diff_lag1
0,01	0,01	0,07	accepted	DXY_1_diff_lag1
0,01	0,01	0,08	accepted	DXY_log_1_diff_lag1
0,01	0,01	0,17	accepted	Ibovespa_1_diff_lag1
0,01	0,01	0,16	accepted	Ibovespa_log_1_diff_lag1
0,01	0,01	0,07	accepted	Brasil_CDS5Y_1_diff_lag1
0,01	0,01	0,07	accepted	Brasil_CDS5Y_log_1_diff_lag1
0,01	0,01	0,09	accepted	crb_1_diff_lag1
0,01	0,01	0,11	accepted	crb_food_1_diff_lag1
0,01	0,01	0,06	accepted	crb_metals_1_diff_lag1
0,01	0,01	0,10	accepted	crb_log_1_diff_lag1
0,01	0,01	0,11	accepted	crb_food_log_1_diff_lag1
0,01	0,01	0,06	accepted	crb_metals_log_1_diff_lag1
0,01	0,01	0,08	accepted	oil_price_brent_1_diff_lag1
0,01	0,01	0,07	accepted	oil_price_wti_1_diff_lag1
0,01	0,01	0,07	accepted	oil_price_brent_log_1_diff_lag1
0,01	0,01	0,06	accepted	oil_price_wti_log_1_diff_lag1
0,01	0,01	0,05	accepted	consu_energia_brasil_total_1_diff_lag1
0,01	0,01	0,08	accepted	consu_energia_industr_1_diff_lag1
0,01	0,01	0,04	accepted	consu_energia_residencia_1_diff_lag1

0,01	0,01	0,03	accepted	consu_energia_comercial_1_diff_lag1
0,01	0,01	0,06	accepted	consu_energia_outros_1_diff_lag1
0,01	0,01	0,04	accepted	consu_energia_brasil_total_log_1_diff_lag1
0,01	0,01	0,07	accepted	consu_energia_industr_log_1_diff_lag1
0,01	0,01	0,03	accepted	consu_energia_residencia_log_1_diff_lag1
0,01	0,01	0,03	accepted	consu_energia_comercial_log_1_diff_lag1
0,01	0,01	0,09	accepted	consu_energia_outros_log_1_diff_lag1
0,01	0,01	0,07	accepted	base_monetaria_1_diff_lag1
0,01	0,01	0,05	accepted	base_monetaria_log_1_diff_lag1
0,37	0,01	0,19	rejected	selic_aa_1_diff_lag1
0,01	0,01	0,21	accepted	consume_confidence_1_diff_lag1
0,01	0,01	0,04	accepted	caged_saldo_norte_1_diff_lag1
0,01	0,01	0,02	accepted	caged_saldo_nordeste_1_diff_lag1
0,01	0,01	0,04	accepted	caged_saldo_sudeste_1_diff_lag1
0,01	0,01	0,05	accepted	caged_saldo_sul_1_diff_lag1
0,01	0,01	0,04	accepted	caged_saldo_centro_oeste_1_diff_lag1
0,01	0,01	0,04	accepted	caged_saldo_brasil_1_diff_lag1
0,01	0,01	0,05	accepted	consu_energia_brasil_total_2_diff_lag1
0,01	0,01	0,08	accepted	consu_energia_industr_2_diff_lag1
0,01	0,01	0,04	accepted	consu_energia_residencia_2_diff_lag1
0,01	0,01	0,03	accepted	consu_energia_comercial_2_diff_lag1
0,01	0,01	0,06	accepted	consu_energia_outros_2_diff_lag1
0,01	0,01	0,04	accepted	consu_energia_brasil_total_log_2_diff_lag1
0,01	0,01	0,07	accepted	consu_energia_industr_log_2_diff_lag1
0,01	0,01	0,03	accepted	consu_energia_residencia_log_2_diff_lag1
0,01	0,01	0,03	accepted	consu_energia_comercial_log_2_diff_lag1
0,01	0,01	0,09	accepted	consu_energia_outros_log_2_diff_lag1
0,01	0,01	0,07	accepted	base_monetaria_2_diff_lag1
0,01	0,01	0,05	accepted	base_monetaria_log_2_diff_lag1
0,37	0,01	0,19	rejected	selic_aa_2_diff_lag1
0,01	0,01	0,30	accepted	PNAD_DESOC_2_diff_lag1
0,01	0,01	0,11	accepted	var_ipca_bacen_lag2
0,08	0,01	0,14	rejected	var_comercializaveis_bacen_lag2
0,01	0,01	0,40	rejected	var_nao_comercializaveis_bacen_lag2
0,01	0,01	0,08	accepted	var_administrados_bacen_lag2
0,01	0,01	0,09	accepted	var_bens_nao_duraveis_bacen_lag2
0,73	0,01	0,29	rejected	var_bens_semi_duraveis_bacen_lag2
0,40	0,01	0,59	rejected	var_duraveis_bacen_lag2
0,01	0,01	0,96	rejected	var_servicos_bacen_lag2
0,27	0,01	0,25	rejected	var_medias_aparadas_sem_suavizacao_bacen_lag2
0,02	0,01	0,34	accepted	var_nucleo_por_exclusao_ex0_bacen_lag2
0,01	0,01	0,19	accepted	var_livres_bacen_lag2
0,04	0,01	0,26	accepted	var_nucleo_por_exclusao_ex1_bacen_lag2

0,21	0,01	0,28	rejected	var_nucleo_dupla_ponderacao_bacen_lag2
0,13	0,01	0,31	rejected	var_difusao_bacen_lag2
0,50	0,01	0,45	rejected	var_nucleo_por_exclusao_ex2_bacen_lag2
0,48	0,01	0,47	rejected	var_nucleo_por_exclusao_ex3_bacen_lag2
0,23	0,01	0,20	rejected	var_industriais_bacen_lag2
0,01	0,01	0,07	accepted	var_alimentacao_em_domicilio_bacen_lag2
0,38	0,01	0,35	rejected	var_nucleo_percentil_55_bacen_lag2
0,03	0,01	0,28	accepted	var_nucleo_ex_alimentacao_energia_exfe_bacen_lag2
0,01	0,01	0,12	accepted	var_incc_lag2
0,02	0,01	2,50	rejected	consu_energia_brasil_total_lag2
0,01	0,01	2,94	rejected	consu_energia_residencia_lag2
0,07	0,01	2,88	rejected	consu_energia_outros_lag2
0,01	0,01	0,73	rejected	caged_saldo_norte_lag2
0,01	0,01	0,43	rejected	caged_saldo_nordeste_lag2
0,02	0,01	0,48	rejected	caged_saldo_sudeste_lag2
0,02	0,01	0,36	rejected	caged_saldo_sul_lag2
0,01	0,01	0,61	rejected	caged_saldo_centro_oeste_lag2
0,02	0,01	0,54	rejected	caged_saldo_brasil_lag2
0,02	0,01	2,49	rejected	consu_energia_brasil_total_log_lag2
0,01	0,01	2,93	rejected	consu_energia_residencia_log_lag2
0,07	0,01	2,82	rejected	consu_energia_outros_log_lag2
0,01	0,01	0,05	accepted	MSCI_emerging_1_diff_lag2
0,01	0,01	0,05	accepted	MSCI_emerging_log_1_diff_lag2
0,01	0,01	0,09	accepted	NOAA_Index_1_diff_lag2
0,22	0,01	0,40	rejected	U.S..Treasury.3.months.nominal.yield_1_diff_lag2
0,04	0,01	0,32	accepted	U.S..Treasury.2.years.nominal.yield_1_diff_lag2
0,01	0,01	0,28	accepted	U.S..Treasury.10.years.nominal.yield_1_diff_lag2
0,01	0,01	0,29	accepted	PNAD_DESOC_1_diff_lag2
0,01	0,01	0,08	accepted	BRL_USD_1_diff_lag2
0,01	0,01	0,10	accepted	BRL_USD_log_1_diff_lag2
0,01	0,01	0,05	accepted	VIX_1_diff_lag2
0,01	0,01	0,06	accepted	VIX_log_1_diff_lag2
0,01	0,01	0,07	accepted	DXY_1_diff_lag2
0,01	0,01	0,08	accepted	DXY_log_1_diff_lag2
0,01	0,01	0,14	accepted	Ibovespa_1_diff_lag2
0,01	0,01	0,14	accepted	Ibovespa_log_1_diff_lag2
0,01	0,01	0,06	accepted	Brasil_CDS5Y_1_diff_lag2
0,01	0,01	0,06	accepted	Brasil_CDS5Y_log_1_diff_lag2
0,01	0,01	0,10	accepted	crb_1_diff_lag2
0,01	0,01	0,14	accepted	crb_food_1_diff_lag2
0,01	0,01	0,05	accepted	crb_metals_1_diff_lag2
0,01	0,01	0,10	accepted	crb_log_1_diff_lag2
0,01	0,01	0,14	accepted	crb_food_log_1_diff_lag2

0,01	0,01	0,06	accepted	crb_metals_log_1_diff_lag2
0,01	0,01	0,09	accepted	oil_price_brent_1_diff_lag2
0,01	0,01	0,08	accepted	oil_price_wti_1_diff_lag2
0,01	0,01	0,07	accepted	oil_price_brent_log_1_diff_lag2
0,01	0,01	0,07	accepted	oil_price_wti_log_1_diff_lag2
0,01	0,01	0,04	accepted	consu_energia_brasil_total_1_diff_lag2
0,01	0,01	0,09	accepted	consu_energia_industr_1_diff_lag2
0,01	0,01	0,03	accepted	consu_energia_residencia_1_diff_lag2
0,01	0,01	0,02	accepted	consu_energia_comercial_1_diff_lag2
0,01	0,01	0,07	accepted	consu_energia_outros_1_diff_lag2
0,01	0,01	0,04	accepted	consu_energia_brasil_total_log_1_diff_lag2
0,01	0,01	0,08	accepted	consu_energia_industr_log_1_diff_lag2
0,01	0,01	0,02	accepted	consu_energia_residencia_log_1_diff_lag2
0,01	0,01	0,03	accepted	consu_energia_comercial_log_1_diff_lag2
0,01	0,01	0,10	accepted	consu_energia_outros_log_1_diff_lag2
0,01	0,01	0,06	accepted	ibc_br_sa_1_diff_lag2
0,01	0,01	0,06	accepted	ibc_br_sa_log_1_diff_lag2
0,01	0,01	0,07	accepted	base_monetaria_1_diff_lag2
0,01	0,01	0,05	accepted	base_monetaria_log_1_diff_lag2
0,39	0,01	0,19	rejected	selic_aa_1_diff_lag2
0,01	0,01	0,17	accepted	consume_confidence_1_diff_lag2
0,01	0,01	0,02	accepted	caged_saldo_norte_1_diff_lag2
0,01	0,01	0,02	accepted	caged_saldo_nordeste_1_diff_lag2
0,01	0,01	0,04	accepted	caged_saldo_sudeste_1_diff_lag2
0,01	0,01	0,02	accepted	caged_saldo_sul_1_diff_lag2
0,01	0,01	0,02	accepted	caged_saldo_centro_oeste_1_diff_lag2
0,01	0,01	0,03	accepted	caged_saldo_brasil_1_diff_lag2
0,01	0,01	0,04	accepted	consu_energia_brasil_total_2_diff_lag2
0,01	0,01	0,09	accepted	consu_energia_industr_2_diff_lag2
0,01	0,01	0,03	accepted	consu_energia_residencia_2_diff_lag2
0,01	0,01	0,02	accepted	consu_energia_comercial_2_diff_lag2
0,01	0,01	0,07	accepted	consu_energia_outros_2_diff_lag2
0,01	0,01	0,04	accepted	consu_energia_brasil_total_log_2_diff_lag2
0,01	0,01	0,08	accepted	consu_energia_industr_log_2_diff_lag2
0,01	0,01	0,02	accepted	consu_energia_residencia_log_2_diff_lag2
0,01	0,01	0,03	accepted	consu_energia_comercial_log_2_diff_lag2
0,01	0,01	0,10	accepted	consu_energia_outros_log_2_diff_lag2
0,01	0,01	0,06	accepted	ibc_br_sa_2_diff_lag2
0,01	0,01	0,06	accepted	ibc_br_sa_log_2_diff_lag2
0,01	0,01	0,07	accepted	base_monetaria_2_diff_lag2
0,01	0,01	0,05	accepted	base_monetaria_log_2_diff_lag2
0,39	0,01	0,19	rejected	selic_aa_2_diff_lag2
0,01	0,01	0,29	accepted	PNAD_DESOC_2_diff_lag2

0,01	0,01	0,11	accepted	var_ipca_bacen_lag3
0,08	0,01	0,15	rejected	var_comercializaveis_bacen_lag3
0,01	0,01	0,41	rejected	var_nao_comercializaveis_bacen_lag3
0,01	0,01	0,08	accepted	var_administrados_bacen_lag3
0,01	0,01	0,09	accepted	var_bens_nao_duraveis_bacen_lag3
0,70	0,01	0,32	rejected	var_bens_semi_duraveis_bacen_lag3
0,29	0,01	0,62	rejected	var_duraveis_bacen_lag3
0,01	0,01	0,99	rejected	var_servicos_bacen_lag3
0,25	0,01	0,25	rejected	var_medias_aparadas_sem_suavizacao_bacen_lag3
0,02	0,01	0,35	rejected	var_nucleo_por_exclusao_ex0_bacen_lag3
0,01	0,01	0,19	accepted	var_livres_bacen_lag3
0,04	0,01	0,27	accepted	var_nucleo_por_exclusao_ex1_bacen_lag3
0,19	0,01	0,29	rejected	var_nucleo_dupla_ponderacao_bacen_lag3
0,13	0,01	0,30	rejected	var_difusao_bacen_lag3
0,47	0,01	0,46	rejected	var_nucleo_por_exclusao_ex2_bacen_lag3
0,46	0,01	0,49	rejected	var_nucleo_por_exclusao_ex3_bacen_lag3
0,14	0,01	0,23	rejected	var_industriais_bacen_lag3
0,01	0,01	0,07	accepted	var_alimentacao_em_domicilio_bacen_lag3
0,36	0,01	0,37	rejected	var_nucleo_percentil_55_bacen_lag3
0,03	0,01	0,28	accepted	var_nucleo_ex_alimentacao_energia_exfe_bacen_lag3
0,01	0,01	0,12	accepted	var_incc_lag3
0,02	0,01	2,49	rejected	consu_energia_brasil_total_lag3
0,01	0,01	2,93	rejected	consu_energia_residencia_lag3
0,17	0,01	2,87	rejected	consu_energia_outros_lag3
0,01	0,01	0,70	rejected	caged_saldo_norte_lag3
0,01	0,01	0,41	rejected	caged_saldo_nordeste_lag3
0,02	0,01	0,46	rejected	caged_saldo_sudeste_lag3
0,02	0,01	0,36	rejected	caged_saldo_sul_lag3
0,01	0,01	0,63	rejected	caged_saldo_centro_oeste_lag3
0,02	0,01	0,52	rejected	caged_saldo_brasil_lag3
0,02	0,01	2,48	rejected	consu_energia_brasil_total_log_lag3
0,01	0,01	2,91	rejected	consu_energia_residencia_log_lag3
0,01	0,01	0,06	accepted	MSCI_emerging_1_diff_lag3
0,01	0,01	0,05	accepted	MSCI_emerging_log_1_diff_lag3
0,01	0,01	0,11	accepted	NOAA_Index_1_diff_lag3
0,22	0,01	0,43	rejected	U.S..Treasury.3.months.nominal.yield_1_diff_lag3
0,05	0,01	0,42	rejected	U.S..Treasury.2.years.nominal.yield_1_diff_lag3
0,01	0,01	0,41	rejected	U.S..Treasury.10.years.nominal.yield_1_diff_lag3
0,01	0,01	0,29	accepted	PNAD_DESOC_1_diff_lag3
0,01	0,01	0,06	accepted	BRL_USD_1_diff_lag3
0,01	0,01	0,09	accepted	BRL_USD_log_1_diff_lag3
0,01	0,01	0,02	accepted	VIX_1_diff_lag3
0,01	0,01	0,03	accepted	VIX_log_1_diff_lag3

0,01	0,01	0,05	accepted	DXY_1_diff_lag3
0,01	0,01	0,05	accepted	DXY_log_1_diff_lag3
0,01	0,01	0,15	accepted	Ibovespa_1_diff_lag3
0,01	0,01	0,17	accepted	Ibovespa_log_1_diff_lag3
0,01	0,01	0,05	accepted	Brasil_CDS5Y_1_diff_lag3
0,01	0,01	0,05	accepted	Brasil_CDS5Y_log_1_diff_lag3
0,01	0,01	0,11	accepted	crb_1_diff_lag3
0,01	0,01	0,17	accepted	crb_food_1_diff_lag3
0,01	0,01	0,05	accepted	crb_metals_1_diff_lag3
0,01	0,01	0,12	accepted	crb_log_1_diff_lag3
0,01	0,01	0,16	accepted	crb_food_log_1_diff_lag3
0,01	0,01	0,06	accepted	crb_metals_log_1_diff_lag3
0,01	0,01	0,10	accepted	oil_price_brent_1_diff_lag3
0,01	0,01	0,09	accepted	oil_price_wti_1_diff_lag3
0,01	0,01	0,08	accepted	oil_price_brent_log_1_diff_lag3
0,01	0,01	0,09	accepted	oil_price_wti_log_1_diff_lag3
0,01	0,01	0,04	accepted	consu_energia_brasil_total_1_diff_lag3
0,01	0,01	0,10	accepted	consu_energia_industr_1_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_residencia_1_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_comercial_1_diff_lag3
0,01	0,01	0,06	accepted	consu_energia_outros_1_diff_lag3
0,01	0,01	0,03	accepted	consu_energia_brasil_total_log_1_diff_lag3
0,01	0,01	0,10	accepted	consu_energia_industr_log_1_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_residencia_log_1_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_comercial_log_1_diff_lag3
0,01	0,01	0,09	accepted	consu_energia_outros_log_1_diff_lag3
0,01	0,01	0,06	accepted	ibc_br_sa_1_diff_lag3
0,01	0,01	0,06	accepted	ibc_br_sa_log_1_diff_lag3
0,01	0,01	0,06	accepted	base_monetaria_1_diff_lag3
0,01	0,01	0,06	accepted	base_monetaria_log_1_diff_lag3
0,38	0,01	0,20	rejected	selic_aa_1_diff_lag3
0,01	0,01	0,18	accepted	consume_confidence_1_diff_lag3
0,01	0,01	0,03	accepted	caged_saldo_norte_1_diff_lag3
0,01	0,01	0,02	accepted	caged_saldo_nordeste_1_diff_lag3
0,01	0,01	0,04	accepted	caged_saldo_sudeste_1_diff_lag3
0,01	0,01	0,03	accepted	caged_saldo_sul_1_diff_lag3
0,01	0,01	0,03	accepted	caged_saldo_centro_oeste_1_diff_lag3
0,01	0,01	0,04	accepted	caged_saldo_brasil_1_diff_lag3
0,01	0,01	0,04	accepted	consu_energia_brasil_total_2_diff_lag3
0,01	0,01	0,10	accepted	consu_energia_industr_2_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_residencia_2_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_comercial_2_diff_lag3
0,01	0,01	0,06	accepted	consu_energia_outros_2_diff_lag3

0,01	0,01	0,03	accepted	consu_energia_brasil_total_log_2_diff_lag3
0,01	0,01	0,10	accepted	consu_energia_industr_log_2_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_residencia_log_2_diff_lag3
0,01	0,01	0,02	accepted	consu_energia_comercial_log_2_diff_lag3
0,01	0,01	0,09	accepted	consu_energia_outros_log_2_diff_lag3
0,01	0,01	0,06	accepted	ibc_br_sa_2_diff_lag3
0,01	0,01	0,06	accepted	ibc_br_sa_log_2_diff_lag3
0,01	0,01	0,06	accepted	base_monetaria_2_diff_lag3
0,01	0,01	0,06	accepted	base_monetaria_log_2_diff_lag3
0,38	0,01	0,20	rejected	selic_aa_2_diff_lag3
0,01	0,01	0,29	accepted	PNAD_DESOC_2_diff_lag3
0,01	0,01	0,11	accepted	var_ipca_bacen_lag4
0,10	0,01	0,15	rejected	var_comercializaveis_bacen_lag4
0,01	0,01	0,43	rejected	var_nao_comercializaveis_bacen_lag4
0,01	0,01	0,08	accepted	var_administrados_bacen_lag4
0,01	0,01	0,09	accepted	var_bens_nao_duraveis_bacen_lag4
0,69	0,01	0,33	rejected	var_bens_semi_duraveis_bacen_lag4
0,28	0,01	0,63	rejected	var_duraveis_bacen_lag4
0,01	0,01	1,00	rejected	var_servicos_bacen_lag4
0,21	0,01	0,25	rejected	var_medias_aparadas_sem_suavizacao_bacen_lag4
0,02	0,01	0,35	rejected	var_nucleo_por_exclusao_ex0_bacen_lag4
0,01	0,01	0,20	accepted	var_livres_bacen_lag4
0,03	0,01	0,26	accepted	var_nucleo_por_exclusao_ex1_bacen_lag4
0,15	0,01	0,28	rejected	var_nucleo_dupla_ponderacao_bacen_lag4
0,14	0,01	0,29	rejected	var_difusao_bacen_lag4
0,46	0,01	0,46	rejected	var_nucleo_por_exclusao_ex2_bacen_lag4
0,44	0,01	0,49	rejected	var_nucleo_por_exclusao_ex3_bacen_lag4
0,16	0,01	0,25	rejected	var_industriais_bacen_lag4
0,01	0,01	0,07	accepted	var_alimentacao_em_domicilio_bacen_lag4
0,33	0,01	0,36	rejected	var_nucleo_percentil_55_bacen_lag4
0,02	0,01	0,28	accepted	var_nucleo_ex_alimentacao_energia_exfe_bacen_lag4
0,01	0,01	0,13	accepted	var_incc_lag4
0,01	0,01	2,46	rejected	consu_energia_brasil_total_lag4
0,01	0,01	2,90	rejected	consu_energia_residencia_lag4
0,31	0,01	2,86	rejected	consu_energia_outros_lag4
0,01	0,01	0,68	rejected	caged_saldo_norte_lag4
0,01	0,01	0,40	rejected	caged_saldo_nordeste_lag4
0,02	0,01	0,44	rejected	caged_saldo_sudeste_lag4
0,02	0,01	0,35	rejected	caged_saldo_sul_lag4
0,01	0,01	0,59	rejected	caged_saldo_centro_oeste_lag4
0,02	0,01	0,49	rejected	caged_saldo_brasil_lag4
0,01	0,01	2,45	rejected	consu_energia_brasil_total_log_lag4
0,01	0,01	2,89	rejected	consu_energia_residencia_log_lag4

0,37	0,02	2,80	rejected	consu_energia_outros_log_lag4
0,01	0,01	0,05	accepted	MSCI_emerging_1_diff_lag4
0,01	0,01	0,04	accepted	MSCI_emerging_log_1_diff_lag4
0,01	0,01	0,11	accepted	NOAA_Index_1_diff_lag4
0,22	0,01	0,43	rejected	U.S..Treasury.3.months.nominal.yield_1_diff_lag4
0,06	0,01	0,43	rejected	U.S..Treasury.2.years.nominal.yield_1_diff_lag4
0,01	0,01	0,38	rejected	U.S..Treasury.10.years.nominal.yield_1_diff_lag4
0,01	0,01	0,28	accepted	PNAD_DESOC_1_diff_lag4
0,01	0,01	0,07	accepted	BRL_USD_1_diff_lag4
0,01	0,01	0,09	accepted	BRL_USD_log_1_diff_lag4
0,01	0,01	0,02	accepted	VIX_1_diff_lag4
0,01	0,01	0,03	accepted	VIX_log_1_diff_lag4
0,01	0,01	0,05	accepted	DXY_1_diff_lag4
0,01	0,01	0,05	accepted	DXY_log_1_diff_lag4
0,01	0,01	0,16	accepted	Ibovespa_1_diff_lag4
0,01	0,01	0,18	accepted	Ibovespa_log_1_diff_lag4
0,01	0,01	0,06	accepted	Brasil_CDS5Y_1_diff_lag4
0,01	0,01	0,06	accepted	Brasil_CDS5Y_log_1_diff_lag4
0,01	0,01	0,12	accepted	crb_1_diff_lag4
0,01	0,01	0,17	accepted	crb_food_1_diff_lag4
0,01	0,01	0,05	accepted	crb_metals_1_diff_lag4
0,01	0,01	0,12	accepted	crb_log_1_diff_lag4
0,01	0,01	0,17	accepted	crb_food_log_1_diff_lag4
0,01	0,01	0,06	accepted	crb_metals_log_1_diff_lag4
0,01	0,01	0,10	accepted	oil_price_brent_1_diff_lag4
0,01	0,01	0,10	accepted	oil_price_wti_1_diff_lag4
0,01	0,01	0,08	accepted	oil_price_brent_log_1_diff_lag4
0,01	0,01	0,09	accepted	oil_price_wti_log_1_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_brasil_total_1_diff_lag4
0,01	0,01	0,08	accepted	consu_energia_industr_1_diff_lag4
0,01	0,01	0,01	accepted	consu_energia_residencia_1_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_comercial_1_diff_lag4
0,01	0,01	0,07	accepted	consu_energia_outros_1_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_brasil_total_log_1_diff_lag4
0,01	0,01	0,08	accepted	consu_energia_industr_log_1_diff_lag4
0,01	0,01	0,01	accepted	consu_energia_residencia_log_1_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_comercial_log_1_diff_lag4
0,01	0,01	0,09	accepted	consu_energia_outros_log_1_diff_lag4
0,01	0,01	0,07	accepted	ibc_br_sa_1_diff_lag4
0,01	0,01	0,06	accepted	ibc_br_sa_log_1_diff_lag4
0,01	0,01	0,06	accepted	base_monetaria_1_diff_lag4
0,01	0,01	0,05	accepted	base_monetaria_log_1_diff_lag4
0,37	0,01	0,21	rejected	selic_aa_1_diff_lag4

0,01	0,01	0,20	accepted	consume_confidence_1_diff_lag4
0,01	0,01	0,03	accepted	caged_saldo_norte_1_diff_lag4
0,01	0,01	0,02	accepted	caged_saldo_nordeste_1_diff_lag4
0,01	0,01	0,04	accepted	caged_saldo_sudeste_1_diff_lag4
0,01	0,01	0,03	accepted	caged_saldo_sul_1_diff_lag4
0,01	0,01	0,02	accepted	caged_saldo_centro_oeste_1_diff_lag4
0,01	0,01	0,03	accepted	caged_saldo_brasil_1_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_brasil_total_2_diff_lag4
0,01	0,01	0,08	accepted	consu_energia_industr_2_diff_lag4
0,01	0,01	0,01	accepted	consu_energia_residencia_2_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_comercial_2_diff_lag4
0,01	0,01	0,07	accepted	consu_energia_outros_2_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_brasil_total_log_2_diff_lag4
0,01	0,01	0,08	accepted	consu_energia_industr_log_2_diff_lag4
0,01	0,01	0,01	accepted	consu_energia_residencia_log_2_diff_lag4
0,01	0,01	0,02	accepted	consu_energia_comercial_log_2_diff_lag4
0,01	0,01	0,09	accepted	consu_energia_outros_log_2_diff_lag4
0,01	0,01	0,07	accepted	ibc_br_sa_2_diff_lag4
0,01	0,01	0,06	accepted	ibc_br_sa_log_2_diff_lag4
0,01	0,01	0,06	accepted	base_monetaria_2_diff_lag4
0,01	0,01	0,05	accepted	base_monetaria_log_2_diff_lag4
0,37	0,01	0,21	rejected	selic_aa_2_diff_lag4
0,01	0,01	0,28	accepted	PNAD_DESOC_2_diff_lag4
0,01	0,01	0,14	accepted	var_ipca_bacen

Note 1: ADF = Augmented Dickey-Fuller, PP = Phillips-Perron, and KPSS = Kwiatkowski-Phillips-Schmidt-Shin

Note 2: P-value < 0.01 suggests stationarity for the ADF and PP tests, while for the KPSS test indicates non-stationarity.

Note 3: The column "Stationarity Decision" means that the series is stationary (accepted) or non-stationary (rejected) based on the result for the three tests.

Appendix B

Average Coefficient Levels of the Selected Features by ElasticNet

Table 11: Average Coefficient Levels

Feature	Coefficient
ipca_monitor_0	0.10
var_ipc_fipe	0.09
ipca_monitor_11	0.09
ipca_focus	0.08
ipca_monitor_1	0.08
consu_energia_residencia_log_1_diff_lag2	0.07
var_ipc_br	0.06
var_ipca_15	0.05
ipca_monitor_5	0.05
BRL_USD_log_1_diff_lag4	0.04
var_livres_bacen_lag1	0.03
consu_energia_comercial_log_1_diff_lag1	0.03
U_S___Treasury_10_years_nominal_yield_1_diff	0.03
consu_energia_outros_log_1_diff_lag4	0.02
caged_saldo_nordeste_1_diff_lag1	0.02
PNAD_DESOC_2_diff_lag1	0.02
consume_confidence_1_diff_lag3	0.02
U_S___Treasury_10_years_nominal_yield_1_diff_lag2	0.01
consu_energia_brasil_total_log_1_diff_lag1	0.01
consu_energia_brasil_total_log_1_diff_lag3	0.00
BRL_USD_log_1_diff_lag3	-0.01
PNAD_DESOC_2_diff_lag3	-0.01
DXY_log_1_diff_lag4	-0.01
NOAA_Index_1_diff	-0.02
BRL_USD_log_1_diff	-0.02
consu_energia_outros_log_1_diff_lag3	-0.02
PNAD_DESOC_2_diff_lag4	-0.03
caged_saldo_nordeste_1_diff_lag3	-0.03
caged_saldo_nordeste_1_diff_lag2	-0.03
var_livres_bacen_lag4	-0.03

Appendix C

Most Frequent Features Selected by ElasticNet

Table 12: Frequency Table.

Feature	Absolute Frequency	Relative Frequency
ipca_monitor_0	108	69.2%
ipca_focus	93	59.6%
BRL_USD_log_1_diff	58	37.2%
consu_energia_outros_log_1_diff_lag4	57	36.5%
PNAD_DESOC_2_diff_lag1	57	36.5%
PNAD_DESOC_2_diff_lag4	56	35.9%
var_ipc_br	55	35.3%
consume_confidence_1_diff_lag3	55	35.3%
ipca_monitor_1	51	32.7%
caged_saldo_nordeste_1_diff_lag2	51	32.7%
consu_energia_brasil_total_log_1_diff_lag3	51	32.7%
consu_energia_outros_log_1_diff_lag3	50	32.1%
U_S___Treasury_10_years_nominal_yield_1_diff	48	30.8%
var_ipc_fipe	48	30.8%
BRL_USD_log_1_diff_lag4	47	30.1%
ipca_monitor_11	47	30.1%
U_S___Treasury_10_years_nominal_yield_1_diff_lag2	47	30.1%
caged_saldo_nordeste_1_diff_lag3	46	29.5%
ipca_monitor_5	46	29.5%
BRL_USD_log_1_diff_lag3	45	28.8%
var_livres_bacen_lag4	45	28.8%
NOAA_Index_1_diff	43	27.6%
consu_energia_residencia_log_1_diff_lag2	43	27.6%
consu_energia_brasil_total_log_1_diff_lag1	43	27.6%
var_livres_bacen_lag1	43	27.6%
DXY_log_1_diff_lag4	42	26.9%
var_ipca_15	42	26.9%
PNAD_DESOC_2_diff_lag3	41	26.3%
caged_saldo_nordeste_1_diff_lag1	41	26.3%
consu_energia_comercial_log_1_diff_lag1	41	26.3%