CHARLES UNIVERSITY
FACULTY OF SOCIAL SCIENCES
Institute of Economic Studies

Machine Learning in Macroeconomic Nowcasting

Master's thesis

Author: Alex Zayat
Study program: Master in Economics and Finance
Supervisor: Prof. PhDr. Ladislav Krištoufek, Ph. D
Year of defense: 2024

Declaration of Authorship

The author hereby declares that he or she compiled this thesis independently, using only the listed resources and literature, and the thesis has not been used to obtain any other academic title.

The author grants to Charles University permission to reproduce and to distribute copies of this thesis in whole or in part and agrees with the thesis being used for study and scientific purposes.

Prague, July 31, 2024

Alex Zayat

Abstract

This study investigates the application of machine learning models for GDP nowcasting, the process of predicting current and near-future economic activity based on high-frequency data. Particularly, the focus is set on their predictive accuracy and interpretability. The performance of various machine learning algorithms, including neural networks, random forests, boosted trees, support vector regression, and K-nearest neighbors, is compared in forecasting Argentina's monthly GDP indicator. The results indicate that machine learning models can enhance predictive accuracy compared to traditional econometric models, aligning with existing literature. Several interpretability techniques are also explored, aiming to understand what insights can be effectively retrieved from these models. It is revealed that the methods are limited in their ability to answer questions related to the functional forms of relationships between variables, but are well-suited to explain the drivers of specific predictions, which is a more important issue in nowcasting. Additionally, a framework for assessing the impact of revisions on predicted estimates is proposed. Ultimately, it is recommended that central banks incorporate machine learning models into their forecasting suites to improve prediction accuracy, while also being mindful of the models' limitations and complexities.

JEL Classification	F12, F21, F23, H25, H71, H87
Keywords	GDP, Nowcasting, Machine Learning, Interpretability, Revisions
Title	Machine Learning in Macroeconomic Nowcasting

Abstrakt

Tento dokument zkoumá aplikaci modelũ strojového učení pro nynějí předpovědi HDP, což je proces předpovídání aktuální a blízké ekonomické aktivity na základě dat s vysokou frekvencí. Zejména je zaměřena pozornost na jejich prediktivní přesnost a interpretovatelnost. Výkon rũzných algoritmũ strojového učení, včetně neuronových sítí, náhodných lesũ, boostovaných stromũ, regresí s podporovýıvektory a K-nejbližších sousedũ, je porovnáván při předpovídání měsíčního ukazatele HDP v Argentině. Výsledky naznačují, že modely strojového učení mohou zlepšit prediktivní přesnost ve srovnání s tradičními

ekonometrický modely, což je v souladu s existujíliteraturou. Také jsou prozkoumány rũzné techniky interpretovatelnosti, které mají za cíl pochopit, jaké poznatky lze efektivně získat z těchto modelũ. Ukazuje se, že metody mají omezenou schopnost odpovíadat na otázky týkajíse funkčních forem vztahũ mezi proměnnými, ale jsou dobře přizpšobeny k vysvětlení faktorũ konkrétních předpovědí, což je v případě nynější předpovědí dũležitější otázka. Kromě toho je navržen rámec pro hodnocení dopadu revizí na předpovězené odhady. Nakonec se doporučuje, aby centrální banky začlenily modely strojového učení do svých předpovědních sad, aby zlepšily prediktivní přesnost, a zároveň by měly být obezřetné vũci omezení mă a složitostem těchto modelũ.

Klasifikace JEL	F12, F21, F23, H25, H71, H87
KlíĂ¨ová slova	GDP, Nynější předpovědi, Strojové učení, Interpretovatelnost, Revize
Název práce	Strojové učení v makroekonomickém nynějším předpovědí

Acknowledgments

I extend sincere gratitude to Prof. PhDr. Ladislav Krištoufek, Ph. D for his support and guidance throughout this research. Special thanks are also due to Luciano Cohan and the Alphacast team for providing full access to their repository, which made the retrieval and processing of the data a seamless experience.

My heartfelt thanks go to my parents and sister, who, from the distance, endured my calls during moments of doubt and frustration; your encouragement helped me persevere and stay focused on my research.

I would like to express my appreciation to Juan Pablo Carranza for his feedback and insightful ideas.

I am also grateful to Samuel Kaplan for first introducing me to the fascinating world of Machine Learning.

Lastly, I would like to extend a particularly special appreciation to everyone at Moody's Analytics. Their support and commitment to my academic journey have significantly enriched my experience and enabled me to successfully complete my Master's studies.

Typeset in LaTeXusing the IES Thesis Template.

Bibliographic Record

Zayat, Alex: *Machine Learning in Macroeconomic Nowcasting.* Master's thesis. Charles University, Faculty of Social Sciences, Institute of Economic Studies, Prague. 2024, pages 76. Advisor: Prof. PhDr. Ladislav Krištoufek, Ph. D

Contents

List of Tables

List of Figures

Acronyms

AR Autoregressive

BT Boosted Trees

EMAE Estimador Mensual de la Actividad Economica

GB Gradient Boosting

GDP Gross Domestic Product

ICC Indice de Confianza del Consumidor

KNN K-Nearest Neighbors

ML Machine Learning

NN Neural Network

RF Random Forest

SHAP Shapley Additive Explanations

SVM Support Vector Machine

SVR Support Vector Regression

XAI Explainable Artificial Intelligence

Master's Thesis Proposal

Author	Alex Zayat
Supervisor	Prof. PhDr. Ladislav Krištoufek, Ph. D
Proposed topic	Machine Learning in Macroeconomic Nowcasting

Motivation The timely estimation of macroeconomic variables, such as Gross Domestic Product (GDP), is of utmost importance to policymakers, investors, and researchers. Accurate predictions in near real-time enable informed decision-making, risk mitigation, and timely responses to economic fluctuations. Traditional econometric models, such as autoregressive integrated moving average (ARIMA) and vector autoregressive (VAR) models, have been widely used for nowcasting. However, with the emergence and increasing popularity of machine learning techniques, there is an opportunity to enhance the accuracy and breadth of macroeconomic nowcasting.

Machine learning techniques offer several advantages over traditional econometric models. They provide greater flexibility in capturing complex nonlinear relationships in macroeconomic data, resulting in improved prediction accuracy. Furthermore, machine learning models can handle high-dimensional datasets and incorporate a wide range of economic indicators and alternative data sources, leading to more comprehensive predictions. Moreover, machine learning techniques can not only predict existing variables but also extract and create new variables that enhance the understanding and forecasting of economic trends.

One drawback from machine learning algorithms lies in the interpretability field. These models often operate as "black boxes," making it difficult to discern the specific features and patterns driving their predictions. This lack of interpretability can limit the ability to provide policymakers, investors, and researchers with meaningful insights into the economic factors influencing GDP fluctuations. Additionally, it may hinder the adoption of ML-based nowcasting methods in environments where model transparency and interpretability are paramount

Hypotheses

Hypothesis #1: Machine learning regression models exhibit superior performance in capturing complex relationships between economic variables compared to traditional econometric models.

Hypothesis #2: The use of larger datasets in the nowcasting process—which can only be managed through the use of machine learning algorithms—enhances prediction accuracy by incorporating additional information beyond the variables used in traditional models.

Hypothesis #3: Ensemble methods, which combine multiple machine learning models, provide the most accurate and robust estimates for macroeconomic nowcasting compared to individual models, thereby improving the reliability of final predictions.

Hypothesis #4: Achieving interpretability in machine learning algorithms is attainable through computational methods, enhancing the transparency of the underlying model drivers.

Methodology The first hypothesis will be tested using a comparative analysis approach. A set of relevant macroeconomic variables will be selected, and various machine learning regression models, such as neural networks, boosted trees, support vector regression (SVR), and k-nearest neighbors (KNN), will be applied to estimate GDP and CPI inflation. These models will be trained and tested on the same dataset, utilizing evaluation metrics such as mean absolute percentage error (MAPE) and root mean squared error (RMSE). Additionally, a benchmark econometric model, namely an AR(1) model, widely regarded as benchmark in nowcasting, will be included for comparison.

To investigate the second hypothesis, the addition of new variables not typically used in the nowcasting literature will be explored, along with the use of machine learning algorithms for feature cleaning. The flexibility of machine learning models will be leveraged to generate new variables that capture insights from the original macroeconomic variables. These synthetic variables will be incorporated into the nowcasting models, and the impact on accuracy will be assessed.

The third hypothesis will be evaluated by constructing an ensemble model that combines predictions from the most accurate and informative individual models identified in the previous steps. Techniques such as averaging or stacking will be employed to form the ensemble. The performance of the ensemble model will be assessed against the individual models and the benchmark econometric model to determine if it offers superior nowcasting accuracy.

To address the fourth hypothesis, various established methods in the realm of machine learning model interpretation will be employed. These methods will be

utilized to comprehend the impact of each variable within the model. By employing these techniques, the study aims to unravel the intricate relationships between the input variables and the predictions made by the machine learning algorithms. Comparative analysis will be conducted across multiple algorithms, ensuring that the identified relationships remain consistent and robust across different modeling approaches. Furthermore, the interpretations derived from these methods will be assessed to ascertain their alignment with established economic theories or intuitive expectations.

Expected Contribution The anticipated contribution of this research lies in its ability to enrich the existing literature on the application of machine learning (ML) methods to macroeconomic nowcasting. The current landscape primarily centers around how ML models outperform traditional models in terms of data fitting. While this has been an essential facet of research in the field, the focus has often sidestepped the equally crucial dimension of model interpretability.

This study aims to augment the existing body of knowledge by providing additional empirical evidence that supports the superiority of ML models in fitting economic data. By doing so, it reinforces the growing consensus that ML models offer more accurate nowcasting predictions. However, what sets this research apart is the secondary facet of interpretability. The utilization of established interpretability methods within these ML models seeks to delve deeper into the relationships between economic variables, unraveling the "black box" nature often associated with ML algorithms. This interpretability element constitutes a distinctive contribution to the literature and is crucial for elucidating how and why these models make predictions.

The intrinsic value of enhanced interpretability extends beyond academic curiosity. It holds the potential to bridge a significant gap in the broader adoption of ML techniques in macroeconomic nowcasting. By shedding light on the inner workings of these models and demonstrating their utility in understanding complex economic relationships, this research can play a pivotal role in bringing down barriers to their acceptance and implementation by policymakers and practitioners. Ultimately, the anticipated contribution of this research is twofold: to affirm the enhanced predictive performance of ML models over traditional methods and to establish the crucial importance of interpretability in further advancing the field of macroeconomic nowcasting.

Outline

1. Introduction: Providing motivation for integrating machine learning into macroeconomic nowcasting and its potential impact on policy formulation and investment decisions.

2. Literature Review: Synthesizing existing research on the use of machine learning techniques in macroeconomic nowcasting and identifying gaps in the literature.

3. Data: Describing the data used, the process of feature selection, and the creation of synthetic variables using machine learning techniques.

4. Methodology: Detailing the various machine learning models employed (neural networks, boosted trees, SVR, KNN) and explaining their application to macroeconomic nowcasting.

5. Results: Presenting the comparison between the machine learning models, the benchmark econometric model, and assessing the importance of each variable in improving predictions.

6. Interpretability: discussion and use of multiple interpretability methods from the field of machine learning to the exercise of GDP nowcasting.

7. Conclusion: Summarizing the key findings, discussing their implications, and suggesting future directions for research in the field of machine learning in macroeconomic nowcasting.

8. References: Listing the cited sources and providing a comprehensive bibliography in the required format.

Core bibliography

Richardson, A., van Florenstein Mulder, T., Vehbi, T. (2021). Nowcasting GDP using machine-learning algorithms: A real-time assessment. In International Journal of Forecasting (Vol. 37, Issue 2, pp. 941–948).

Banbura, M., Giannone, D., Modugno, M., Reichlin, L. (2013). Now-Casting and the Real-Time Data Flow. In Handbook of Economic Forecasting (pp. 195–237). Elsevier.

Kitchen, J. and R. Monaco (2003). Real-time forecasting in practice: The U.S. Treasury staff's real-time GDP forecast system. Business Economics 38(4), 10-19.

Giannone, D., L. Reichlin, and D. Small (2008). Nowcasting: the real-time informational content of macroeconomic data. Journal of Monetary Economics, 55(4), 665-676

Loermann, J., Maas, B. (2019). Nowcasting US GDP with artificial neural networks. Munich personal repec archive working paper no. 95459.

Muchisha, N. D., Tamara, N., Andriansyah, A., Soleh, A. M. (2021). Nowcasting Indonesia's GDP Growth Using Machine Learning Algorithms. Indonesian Journal of Statistics and Its Applications, 5(2), 355–368

Molnar, Christoph. Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. Leanpub. 2020.

Chapter 1

Introduction

Nowcasting is defined by Banbura *et al.* (2013) as "the prediction of the present, the very near future, and the recent past." This term, a contraction of *forecasting* and *now*, has become a regular exercise mainly among central banks. Nowcasting aims to provide an estimate for current economic conditions, as official indicators, such as GDP, are typically reported with a significant delay. Since timely data is a necessary condition for well-informed decision making—both by policymakers and private agents—these delays constitute a significant drawback on the usefulness of the indicators, and the need for producing early estimates becomes crucial.

The fundamental premise of nowcasting is that certain indicators provide sufficient information to estimate GDP figures prior to their official release. For these indicators to be effective, they must exhibit a strong correlation with GDP and be available well in advance of the official GDP data. Typically, these variables encompass a range of financial metrics, both historical and current macroeconomic data, sector-specific production indicators, and survey results, among others.

Nowcasting is primarily utilized by central banks to support informed decision-making regarding monetary policy and economic intervention. Additionally, other agents may employ nowcasting techniques for different purposes: economic analysts for public disclosure, firms for guiding their strategic decisions, etc. Its relevance in theoretical economic research, on the other hand, is somewhat limited. Where the focus in economic research tends to be on establishing causal relationships, nowcasting's main goal is to accurately predict indicators and understanding the predictions.

Nowcasting encompasses more than just the generation of predictions. Ban-

bura *et al.* (2013) argue that "the nowcasting process goes beyond the simple production of an early estimate [...]" to emphasize the importance of understanding how new data influences these estimates. I align with their perspective and argue that understanding the inner workings of the model, identifying the drivers of the predictions and assessing the impact of data revisions are all integral aspects of the nowcasting process.

Central banks typically rely on a variety of models for their GDP predictions. The most commonly used models include dynamic factor models, bridge equations, and various forms of autoregressive models (AR models), such as ARIMA. Additionally, structural econometric models and Bayesian vector autoregression (BVAR) are often employed.

Machine learning models, however, are primarily designed as prediction tools and offer several advantages over traditional econometric approaches. They can effectively handle larger datasets and are capable of capturing complex, potentially non-linear relationships within the data. With the growing availability of high-frequency data and advancements in computational power, machine learning models are becoming increasingly appealing, especially when the primary goal is to enhance predictive accuracy rather than to conduct causal analysis.

There is a growing body of literature exploring the application of machine learning algorithms for GDP nowcasting. Most of the studies focus on comparing the predictive accuracy of these models to that of traditional economic models, and there is a consensus that machine learning algorithms can significantly improve forecasting accuracy. While central banks are beginning to incorporate machine learning techniques into their modeling suites, several challenges hinder their broader adoption. One major limitation is the lack of transparency associated with these algorithms, which raises concerns about accountability in the predictions they produce. As a result, interpretability becomes a critical factor that restricts the widespread use of machine learning models in GDP nowcasting, as decision-makers often require clear insights into the underlying mechanics of the models to trust their outputs.

This study seeks to contribute to the existing literature in two ways. First, it aims to provide additional evidence regarding the ability of machine learning algorithms to produce more accurate predictions compared to standard econometric models. Second, it tries to bring interpretability into the conversation by demonstrating the application of various methods to elucidate the insights that can be retrieved from thes models and what limitations exist. There are

two smaller chapters discussing the advantages and disadvantages of machine learning algorithms and common preprocessing techniques that are popular in the field for realizing these algorithms' full potential. The ultimate goal is to promote the adoption of machine learning models in economic forecasting by advancing the discussion into new territory.

In this study, the performance of a suite of machine learning algorithms—including a neural network, random forest, boosted trees, support vector regression, and K-nearest neighbors (KNN)—is tested for predicting Argentina's monthly GDP indicator. Additionally, the performances are compared against a benchmark AR(1) model to assess if they are indeed capable of outperforming it. Following the performance assessment, the study explores interpretability techniques that allow for a deeper understanding of how the predictions were generated. Lastly, a simple simulation-based framework is proposed to evaluate the impact of revisions on the predicted estimates, offering insights into the uncertainty surrounding the forecasts.

The remainder of this document is structured as follows: Chapter 3 explores the advantages and disadvantages of machine learning models in the context of regression tasks. Chapter 2 provides a summary of the existing literature on the application of these models for GDP nowcasting. In Chapter 4, various preprocessing techniques commonly used in machine learning are discussed in detail. Chapter 5 introduces the models employed in the empirical study, followed by a concise overview of the data in Chapter 6. Chapter 7 focuses on the performance evaluation process and presents the results of applying the models to the dataset. Chapter 8 emphasizes the significance of interpretability and outlines several techniques available for achieving it. The final analytical section is presented in Chapter 9, where a framework for understanding the impact of revisions in the explanatory variables on the predictions is proposed. Finally, Chapter 10 wraps up the discussion and highlights the key findings of the study.

Chapter 2

Literature Review

Much like the topic of GDP nowcasting grew rapidly among academic studies in the past decade, it is now the application of machine learning techniques for nowcasting that is rapidly evolving. Characterized by a growing body of literature, it is a relatively new topic, as most of the studies in the field appeared in the last 5 years.

Most studies emphasize the ability of machine learning models to fit the data more effectively than traditional econometric approaches, while a smaller segment addresses the interpretability of these models. While explainable machine learning—popularly referred to as XAI—also is a growing topic that has picked interest recently, the literature exploring its implications for economic modeling remains limited.

2.1 Model performance

Several studies emphasize the comparison of fit between different models and against traditional econometric models.

For instance, Richardson *et al.* (2018) assess the performance of a suite of machine learning algorithms including K-Nearest Neighbors (KNN), Least-Squares Boosting, Lasso, Ridge, Elastic Net, Support Vector Machine (SVM), and feedforward Neural Networks for GDP nowcasting in New Zealand. They benchmarked these models' performance with traditional econometric models such as AR(1), a factor model, and Bayesian VAR and find that machine learning algorithms consistently outperform these traditional models, particularly when individual nowcasts are combined. Similarly, Fornaro & Luomaranta (2020) evaluate multiple methods including boosting and regression

trees against an ARIMA benchmark for Finnish economic activity, concluding that combined models provide better fit than both ARIMA and individual methods. Tiffin (2016) also applies elastic net regression and random forests for nowcasting GDP in Lebanon but does not compare against other models, focusing solely on improving fit with ensemble methods. On the U.S. data, studies by Babii *et al.* (2022) and Soybilgen & Yazgan (2021) utilize dynamic factor models and tree-based ensemble models, respectively, and support the conclusion that machine learning models can outperform standard models. Additional studies on U.S. data include Loermann & Maas (2019) and Hopp (2024).

Additionally, there are several studies exploring the effectiveness of machine learning approaches in other contries. Zhang *et al.* (2023) analyze Chinese GDP nowcasting using a wide array of ML models, including ridge regression and dynamic factor models, finding that some machine learning methods surpass the benchmark dynamic factor model. In the case of India, both Ranjan & Ghosh (2021) and Malik & Agarwal (2022) incorporate high-frequency macroeconomic and financial data, with their findings suggesting that machine learning models provide substantial improvements over traditional models. Meanwhile, Jonsson (2020) evaluates a nearest neighbor algorithm for Swedish GDP nowcasting, reporting results comparable to conventional linear indicator models. Other studies include Marcellino & Sivec (2021) for Luxemburg, Tamara et al. (2020) for Indonesia, Dauphin *et al.* (2022) for a sample of European countries, Cepni *et al.* (2019) for emerging countries, and Fan (2019) also for New Zealand.

To avoid redundancy, not every study is discussed in detail. In general, these studies evaluate the fit of a consistent set of standard machine learning algorithms, including Ridge Regression, LASSO, Elastic Net, Random Forest, Boosted Trees, K-Nearest Neighbors, Support Vector Machines, and Neural Networks. The benchmark models primarily consist of AR(1) models, while some studies incorporate traditional econometric approaches such as dynamic factor models, VAR models, or internal model suites utilized by central banks. Overall, the majority of these studies support the conclusion that machine learning algorithms achieve greater prediction accuracy compared to traditional econometric models when applied to the same datasets. Additionally, they demonstrate that ensemble methods can further enhance predictive accuracy.

On the data used

Most studies leverage the capacity of machine learning algorithms to handle larger datasets, incorporating extensive sets of high-frequency data, often with uneven frequencies. Typically, the datasets utilized in these studies comprise macroeconomic and financial variables, with some incorporating additional sources such as survey data, uncertainty indices, text-based variables, and microeconomic data.

Certain studies explicitly address data-related challenges. For instance, Soybilgen & Yazgan (2021) employ dynamic factor models to resolve the ragged edge problem across ten groups of variables. Similarly, Cepni *et al.* (2019) evaluate the performance of their models using various dimensionality reduction techniques to enhance their predictive capabilities.

A noteworthy characteristic of many of these studies is the relatively short timespan of the datasets, which often consist of quarterly data covering periods of 8 to 15 years. This is relevant as machine learning models generally require larger datasets in order to effectively learn from the data.

2.2 Interpretability

Interpretability of machine learning within economic contexts remains mainly uncovered territory, as fewer studies delve into this aspect. Two studies involving GDP nowcasting touch on the topic of interpretbilit: Park & Yang (2022) assess the use of XAI on long short-term memory (LSTM) networks to identify the most important features in the model; then, in their nowcasting study of Egypt's GDP, Abd El-Aal *et al.* (2023) also assess feature importance in their trained random forests and gradient boosting models.

Overall, the available literature underscores the potential of machine learning techniques for improving GDP nowcasting accuracy across various countries and methodologies. The exploration of interpretability, still in its infancy in macroeconomic modeling, will be essential for building trust and understanding the implications of machine learning applications in this field.

Chapter 3

Machine Learning vs. Traditional Econometric Models

Machine learning models are generally considered to offer multiple advantages over traditional econometric models, primarily in their ability to detect non-linear relationships within economic data. Traditional models often rely on linear assumptions and predefined equations, which can limit their ability to capture the full complexity of economic interactions. In contrast, machine learning algorithms are designed to learn from data without explicit programming, enabling them to automatically identify and model complex, non-linear patterns and interactions that may be difficult to capture using conventional methods. This capability is especially valuable in economic modeling, where subtle, non-linear interactions between variables can provide additional insights into GDP dynamics.

The flexibility and adaptability of machine learning algorithms further enhance their appeal. These models can easily adjust to new data and incorporate evolving economic conditions, making them particularly well-suited for forecasting and nowcasting in volatile or uncertain environments. For instance, when new economic indicators or unexpected events arise, machine learning models can quickly adapt by retraining on the updated data. Traditional econometric models, on the other hand, often require significant manual adjustments and recalibration to respond to changes in the data environment, potentially leading to delays and inaccuracies.

Another advantage of machine learning is its proficiency in handling large and diverse datasets. In modern economies, where vast amounts of information are generated daily, the ability to process and analyze big data becomes

increasingly valuable. Modern datasets bundle standard economic indicators, such as sectoral production indicators and financial variables, with newer types of data like text-based indicators such as sentiment measures, online searches and other social media behavior, high-frequency environmental variables, etc. Machine learning models can efficiently manage and extract more information from these extensive datasets, capturing the nuances of economic activities that might be overlooked by traditional methods.

Lastly, and connected to the point above, lie the often built-in mechanisms for feature selection and dimensionality reduction. These techniques, which are essential for identifying the most relevant economic indicators to bring out the information in the data, have the additional effect of reducing the risk of multicollinearity, a common issue in traditional econometric models. Feature selection algorithms, such as LASSO or tree-based methods, can automatically identify and retain the most informative features while discarding irrelevant or redundant ones. Dimensionality reduction techniques, like Principal Component Analysis (PCA) or its more flexible alternative, autoencoders, help simplify the data by reducing the number of variables, thus enhancing predictive performance and providing clearer insights into the factors driving economic changes.

Despite their numerous advantages, machine learning models also present several challenges compared to traditional econometric approaches. One major drawback is the "black box" nature of many machine learning models. These models pass data through multiple layers of non-linear transformations and interactions, making it difficult to interpret the exact pathways and mechanisms through which they generate predictions. Additionally, machine learning algorithms are often part of a broader modeling framework that is preceded by feature selection and dimensionality reduction techniques as outlined above, standardization, etc. These steps add extra layers of complexity to the overall modeling process, further decreasing interpretability. This opacity can obscure the understanding of how specific input features influence the output, raising concerns about accountability and transparency in decision-making processes. For economic policy and forecasting, where interpretability is crucial, this lack of transparency can hinder the acceptance and trust in machine learning-based predictions.

Another significant disadvantage is the susceptibility of machine learning models to overfitting. Without proper regularization and validation, these models may perform exceptionally well on training data but fail to general-

ize effectively to new, unseen data. This issue underscores the importance of rigorous model validation, including techniques such as cross-validation and the use of regularization parameters (e.g., dropout in neural networks or pruning in decision trees) to prevent overfitting and ensure robust nowcasting results. These considerations, while not absent in traditional econometric models, are generally less important as models tend to be simpler and do not require as many validation steps.

Generally speaking, the computational complexity of advanced machine learning models can also pose a challenge. Models such as deep learning architectures, which involve numerous layers and potentially millions of parameters, can be computationally intensive, requiring substantial computing resources and specialized hardware such as GPUs or TPUs. This complexity can lead to longer training times and increased operational costs. While technological advancements and the availability of cloud computing have significantly lowered this barrier, computational complexity remains a consideration, particularly for researchers and institutions with limited resources. However, it is worth noting that for GDP nowcasting, which typically involves a relatively small number of features compared to other applications like image or speech recognition, the computational demands are generally manageable and do not constitute a major constraint.

In summary, while machine learning models offer powerful tools for economic nowcasting, their application requires careful consideration of their limitations. Addressing issues related to interpretability, overfitting, and computational complexity is essential to fully capture the potential of these models and ensure their effective integration into economic forecasting and nowcasting practices.

Chapter 4

Preprocessing techniques

The use of machine learning in economic modeling is not limited to better regression algorithms. Machine learning-based techniques can also be implemented in the preprocessing stage, when large and diverse datasets are being analyzed and transformed for the specific exercise the modeler is doing. While machine learning and traditional econometric models share many preprocessing steps, such as data collection, integration and cleaning, exploratory analysis, or handling of imbalanced data, there are others in which machine learning can provide additional value to the overall modeling pipeline. In this section, the focus is on feature selection and dimensionality reduction techniques, generally considered to be the two steps that most influence the performance and accuracy of predictive models.

Feature selection is the process of identifying and retaining the most relevant variables from a dataset while discarding those that are redundant or irrelevant. This technique is crucial for mitigating multicollinearity, enhancing model interpretability, and improving predictive performance. By focusing on the most informative features, machine learning models can achieve more accurate and robust nowcasting results. Various methods, such as filter-based approaches, wrapper methods, and embedded techniques, can be employed to perform feature selection, each offering distinct advantages and applications.

Dimensionality reduction, on the other hand, involves transforming the original set of variables into a lower-dimensional space while preserving as much of the underlying information as possible. This technique is particularly useful in handling high-dimensional datasets, where the presence of numerous variables can lead to increased computational complexity and overfitting. Dimensionality reduction methods, such as Principal Component Analysis (PCA),

t-Distributed Stochastic Neighbor Embedding (t-SNE) and autoencoders, help simplify the data structure, making it easier to visualize, analyze, and use in the modeling stage.

In the following subsections, various methodologies and machine learning algorithms used for feature selection and dimensionality reduction will be described. We will examine their applications in economic nowcasting, discussing their advantages and limitations. Additionally, in Chapter 7, the impact of a number of these techniques on the predictive accuracy of the models is tested.

4.1 Feature Selection

Feature selection is the process of choosing the subset of explanatory variables or *features* that offer the most predictive power. This process is specially valuable when dealing with larger sets of features or when there are many closely related variables, as it replaces the need for the modeler's judgement on what variables are useful with automatic techniques, helps reducing model complexity, lowers the model's computational costs, reduces the risk of overfitting, and potentially improves the model's performance. The main feature selection techniques can be broadly categorized into three groups: filter methods, wrapper methods, and embedded methods. Only the first two are explained in this section, as embedded models can be addressed directly in the section where regression algorithms are covered (as their name suggests, these methods are incorporated as steps in the algorithm's training process).

4.1.1 Filter Methods

In filter methods, features are usually scored through a statistical test or based on summary statistics. Thus, these methods are independent of the regression or classification algorithm that will be used in a later stage, allowing them to be used in a broad set of applications. In the context of economic modeling, this proves valuable when multiple algorithms are to be run, compared, and potentially integrated.

ANOVA F-value

The simplest filtering method is the ANOVA F-value, where features are ranked based on their individual linear relationship with the outcome variable. This method, which is based on the traditional analysis of variance, is widely used

in economic modeling as it provides a simple implementation and is relatively easy to interpret, at the expense of lost information, since the model only captures linear relationships, potentially defeating the purpose of using Machine Learning algorithms over econometric models.

Mutual Information (MI)

A similar, more powerful technique for identifying the variables with the most explanatory power in a dataset relies on the *Mutual Information* measure. Also called *Information Gain* in other applications, this metric calculates the degree of information one variable can offer about another. Calculated as the difference between unconditional and conditoinal entropy of a given variable:

$$I(X;Y) = H(X)\hat{a}\pounds\imath H(X|Y)$$

where X and Y are two random variables, it allows for the capturing of non-linear relationships as well as linear relationships, thus providing added value on the above-mentioned ANOVA F-value technique.

Variance Threshold

An alternative to statistical tests is relying on descriptive statistics of a series to determine its usefulness. One such technique is filtering variables based on whether or not their variance are above a minimum threshold. The underlying idea is that variables with little variation in their values are unlikely to have explanatory power. At the extreme, a feature with zero variance (i.e., a constant) will have no correlation with any other variable. Outside of the extreme, however, even a low-variance variable can correlate with the outcome variable, and this method fails to take that relationship into account.

Alternatives to using variance are the Mean Absolute Difference (MAD) and Dispersion Ratios.

4.1.2 Wrapper Methods

While filtering methods are agnostic of the regression or classification algorithm to be used and rely on general information about the features, wrapper methods do not analyze the characteristics of the variables, but rather exhaustively test the performance of models with different subsets of features to find

the best possible subset. While filtering methods are naturally more computationally efficient and generally less prone to implementation mistakes, wrapper methods are agnostic of assumptions of specific relationships between variables and provide a fuller picture of the value of the different features.

Naturally, since different models process the information in different ways, the choice of method becomes increasingly important when using wrapper methods, as the selection of variables for one model will not necessarily be optimal for another model.

Exhaustive Feature Selection

The most robust feature selection method tests all possible combinations of features for a given model, and outputs the best performing one. Naturally the most computationally costly because of the number of iterations it requires to find the final result, is usually not chosen, and instead other, less exhaustive methods are considered. It should be noted, however, that when dealing with relatively small datasets, like in the case of economic modeling (with the exception of exercises involving financial markets data, which tend to be larger), this cost is not as important and this method remains a feasible and sensible alternative.

Forward Feature Selection

This iterative method begins with one variable and tests how adding each possible value to the first one improves the model performance to select the best one and incorporate it into the set of features to keep. After a new feature is chosen to be added, the process is repeated on top of the already selected set until a certain criterion is met.

Backward Feature Elimination

This method works in the opposite way to the forward feature selection. When performing backward feature elimination, the starting point is a model including all features, and removing one at a time to find the one that adds the least predictive power. Once one feature is chosen to be removed, the process is repeated with the remaining variables in the model until a certain criterion is met.

Recursive Feature Elimination

Starting from the full set of variables, this method relies on an assessment of importance to select the least contributing feature and remove it from the sample. Once a feature has been permanently removed, the model is retrained and the process is repeated until a certain criterion is met.

Genetic Algorithms

Genetic algorithms are optimization methods inspired by the dynamics of natural selection. In feature selection, these models start with a *population* of feature subsets, evaluate their performance on the machine learning model (the performance metric is generally referred to as *fitness function* in genetic algorithms), and iteratively select the best-performing subsets. The selected variables from different subsets are then combined to create new subsets, and again the best-performing one is selected (this process is called *crossover*). This process is repeated, aiming to find the optimal subset of features that maximizes model performance.

4.2 Dimensionality Reduction

Dimensionality reduction is the practice that aims to simplifying datasets by reducing the number of variables while retaining as much information as possible. The dimensionality reduction method par excellence is Principal Component Analysis (PCA), whose use has expanded across different applications. However, alternative methods have emerged in recent years that provide different advantages to PCA. In this subsection, Kernel PCA (an augmented version of PCA), and autoencoders (neural networks used for dimensionality reduction) are outlined.

4.2.1 Principal Component Analysis (PCA)

Principal Component Analysis is a widely used dimensionality reduction technique that transforms a dataset with potentially correlated features into a set of linearly uncorrelated components called principal components. This transformation is achieved by identifying the directions, or axes, along which the variance in the data is maximized. The first principal component accounts for the greatest variance, followed by the second, and so on, with each sub-

sequent component being orthogonal to the previous ones. By projecting the original data onto these principal components, PCA reduces the number of features while preserving as much variability as possible. This technique relies on simple mathematical relationships, allowing easier traceability in interpreting the construction of the components. While advantageous from a computational efficiency and traceability standpoints, the transformation of the data into principal components obscures the way each initial variable affects the model; even when combined with simple algorithms such as linear regression, the use of PCA implies that the original variables will not have a direct coefficient that can be used for economic analysis. An additional cost of PCA is that it is designed to capture only linear relationships between the variables, potentially leaving information on the table. This lack justifies the advent of alternative methods for dimensionality reduction such as the ones outlined in this section.

4.2.2 Kernel PCA

Kernel Principal Component Analysis is an extension of the standard PCA that allows for capturing non-linear relationships in the data. By means of a kernel function, this method projects the data into a higher-dimensional space to then reduce it again in a similar manner to PCA. This provides additional information to the new set of features, at the expense of reduced interpretability.

4.2.3 Autoencoders

Autoencoders are a type of neural network primarily used for dimensionality reduction. An autoencoder maps the input data to a lower-dimensional latent space and then reconstructs the original data from this compressed representation. The network is trained to minimize the reconstruction error, ensuring that the latent space captures the most significant features of the input data. Neural networks are explained in more detail in Chapter 5, and only the essentials of autoencoders are addressed here.

Starting from the full set of features, the *input layer*, the algorithm performs non-linear transformations on the variables to create new features or *nodes* in a lower-dimensional space. Once the new nodes have been created, they constitute a new layer, and the transformation process is performed again onto a new lower-dimensional set. The overall process is repeated until the desired number of nodes remains. This comprises the first part of the autoencoder,

called the *encoder*. The final result of the encoder is a set of new features that will be the starting point for the second part, the *decoder*.

The decoder acts in the opposite way: it takes the new features from the encoder and, through the use of more transformations, it projects the information of these nodes on a higher-dimensional space. The process works by increasing the dimension in each layer, until the final number of features is achieved in the *output layer*.

Advantages and disadvantages of autoencoders

Autoencoders work with highly non-linear transformations, allowing them to capture more complex relationships than the standard PCA method, constituting one of the main justifications for their use. Additionally, their end use-agnostic nature allows autoencoders to be used in a broad range of applications generally and in combinations with different regression algorithms in particular. Lastly, neural networks are generally flexible, allowing the modeler to customize their network for the specific needs arising from their exercise and data.

The benefits of a more powerful algorithm with increased flexibility naturally comes with an associated computational cost. Neural networks are computationally intensive, with high-dimensional datasets entailing an increased cost in both time and resources. This cost is proportional to the gain of using autoencoders, as more data generally bring out the most benefits from them.

Chapter 5

Models

In this section, an overview of the models utilized in the empirical application will be presented. Since these models are well-established in the literature, the descriptions will be concise, focusing on their key characteristics and relevance to the study. For readers seeking more in-depth information, it is recommended to consult additional resources or relevant literature.

5.1 Neural Networks

Artificial Neural Networks (ANN) are one of the most popular algorithms in machine learning. Inspired by the way the human brain works when processing information, it is used in a broad range of applications such as time series modeling, pattern recognition, generative AI, image recognition, among others. In the economics and finance field, neural networks have witnessed an increase in popularity in recent years, due to their ability to incorporate large and complex datasets that characterize modern economies and financial markets, and their potential to capture more complex dynamics, improving prediction accuracy.

Neural networks consist of interconnected layers of nodes, or *neurons*, that process information in a hierarchical manner. Each node in a neural network is analogous to a biological neuron, receiving input, processing it, and passing the output to the next layer of nodes.[1]

The starting point of a neural network is the *input layer*, comprised by the initial features in the model, which take the name of nodes (x). The nodes

[1]There are several types of ANNs. In this study, only one type is considered, the so called *Feedforward Neural Network*. In particular, a Multilayer Perceptron is described.

in the input layer serve as the input of the second layer, or the first *hidden layer*. Each node in the hidden layer takes the inputs from the previous layer together with a set of *weights* (w) and a *bias* (b), and applies a transformation on them through a non-linear function, the *activation function*, to produce the final value of the node:

$$x_i^{j+1} = f\left(\sum_{i=1}^{n} w_i^{j,j+1} x_i^j + b\right) \tag{5.1}$$

This process is repeated sequentially from one layer to the next, until the last one, the *output layer*, is reached. In regression tasks, the output layer is generally comprised of a single node, which represents the network's prediction of the target value.

The weights and biases in each layer and node of the neural network are initially random, and it is through their adjustment that the network improves its prediction. Neural networks are trained using a process called backpropagation. This involves adjusting the weights based on the difference between the predicted output and the actual target output. The most common algorithm used to minimize the error is *stochastic gradient descent* (SGD).

$$w^{k+1} = w^k - \eta \frac{\partial e}{\partial w^k} \tag{5.2}$$

where η is the learning rate, and $\frac{\partial e}{\partial w^k}$ is the gradient of the error with respect to the weight.

Advantages and disadvantages of Neural Networks

The main advantage of neural networks against other algorithms, and most notably against traditional econometric models, lies in their flexibility and scalability. Neural networks are able to capture non-linear relationships between variables that are potentially lost in other algorithms. Cybenko, G. (1989) proved that certain types of neural networks are able to approximate any function to any desired level of accuracy. This result, popularly referred to as the Universal Approximation Theorem, suggests that neural networks should in theory be able to capture any possible relationship between variables.

The complex nature of neural networks, which results in their flexibility and overall potential, presents challenges and drawbacks. One of the main challenges lies in the data requirements: in order to be trained effectively, neural networks usually need large datasets, which are not always available in

economic modeling exercises (and particularly in developing economies, where the need for better performing algorithms is arguably highest).

An additional challenge, and particular focus of this research, is the lack of transparency of the algorithm. The architecture of neural networks, from the number of parameters (weights and biases) and different layers, to the activation functions used in each transformation, make it increasingly difficult to understand the inner workings of the algorithm and to trace how one variable affects the final prediction. This is a particularly relevant challenge, as interpretability is a fundamental aspect for model selection in economics, where the final prediction is often not the most important output of the exercise, and instead understanding the role of one variable, or identifying the main drivers for a change in output, is of interest to decision makers.

5.2 Regression Trees

Decision trees (and their regression-tailored version, *regression trees*), are another group of popular algorithms in machine learning. When applied to economics and finance, these algorithms are often used for tasks such as credit risk modeling, where borrowers have to be classified into groups based on the probability of them defaulting on their loans, or fraud detection in credit card systems, where transactions have to classified into fraudulent or legitimate according to certain characteristics. Regression trees, however, can also be used for time series modeling, and are particularly useful when combined with explanatory variables whose movements are expected to drive the target.

The process begins with the dataset including the input features and a target variable. The tree-building process involves recursively splitting the data into subsets, based on the value of a given feature (e.g., a numerical variable being above or below a certain threshold, or a categorical variable falling into one group or another). Each split is chosen to maximize the Informational Gain (IG), which is a measure of the increase of *purity* of the nodes after the split. The IG can be defined as follows:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

Here, f is the feature on which the split will be performed, D_p and D_j are the dataset of the parent and j-th child node; I is the impurity measure; N_p and N_j

are the number of observations in the parent and j-th child node, respectively. The IG then measures how much the impurity changes between the parent and the children nodes - a larger IG means a lower impurity of the child nodes, a positive outcome.

The impurity, a measure for the homogeneity in a node, can take different forms depending on the problem at hand. In the case of classification problems, Gini impurity (I_G), entropy (I_H), and classification error (I_E) are commonly used; in regression problems, the mean squared error (MSE) is the usual choice:

$$I(t) = MSE(t) = \frac{1}{N_t} \sum_{i \in D_t} \left(y^{(i)} - \widehat{y}_t \right)^2$$

Here, N_t is the number of observations at node t, $y^{(i)}$ is the true target value, and $\widehat{y}_t$ is the predicted target value (usually the sample mean of the node). In order to choose the right split, the algorithm tests every possible binary split across all available variables and compares their associated IG. For computational reasons, the split considered for each variable is done based on its mean.

It can be seen how the splitting process aims at reducing the variance of the estimation; at the limit, a deep-enough tree can split the N observations into N nodes of $N_t = 1 \forall t$, predicting each observation in the training set perfectly, at the cost of generalization power. Hence, it is common practice to *prune* the trees or stop the optimization process at a certain depth (number of splits) to avoid overfitting. The pruning process involves separating the initial training set into two new sets, namely *training* and *validation*. After the tree is grown to its largest possible on the new training set, the validation set is used in a backwards manner to assess the IG in reverse, to test if a specific split should be removed, as the impurity is smaller on the parent node than in the leaf nodes.

Advantages and disadvantages of Regression Trees

Despite their simple mechanics, regression trees have proved to perform well in regression tasks, often outperforming other more complex algorithms. Trees are able to capture relatively more complex relationships as the splits do not rely on linearity. The main advantage of tree-based algorithms, however, lies in their relatively high interpretability: binary splits are easy to visualize, and the purity-maximizing nature of the splits provides an implicit feature importance

measure (for this reason, trees are also often used as a preprocessing step for feature selection).

The primary disadvantage of using regression trees over other algorithms is that not all variables are used in the splits, hence leaving potentially useful information on the table. An additional drawback of regression trees is their instability, as small changes in the data can lead to a different choice of splits, potentially resulting in severely different trees. This challenges are overcome, however, by the use of *Random Forests*, an ensemble method that builds on multiple trees that are trained on different subsets of the data and potentially using different variables, to then aggregate the information into a final prediction. These algorithms are generally more stable and can incorporate more information as each tree is train individually and together they can make use of a larger set of features. Random forests are briefly described in the following subsection.

5.3 Random Forest

Random forests are an ensemble learning method that builds multiple decision trees during training and outputs the mode of the classes for classification problems or the average prediction for regression problems. The idea is to combine different decision trees that usually suffer from high variance to build a more robust model that is less susceptible to overfitting.

Each tree in the random forest is trained on a *bootstrapped* sample, which is a random sample with replacement from the original dataset. Additionally, at each split in the decision tree, only a random subset of features is considered. This introduces further randomness into the algorithm and adds diversity among the trees, helping prevent overfitting to improve the model's out of sample performance.

For classification tasks, the final prediction is determined by a majority vote among the trees. For regression tasks, it is the average of the predictions.

Classification

$$\text{Final Prediction (Classification)} = \text{Mode}(\text{Predictions from Trees}) \tag{5.3}$$

Regression

$$\text{Final Prediction (Regression)} = \frac{1}{N}\sum_{i=1}^{N}\text{Prediction}_i \tag{5.4}$$

where N is the number of trees and Prediction_i is the prediction of the i-th tree.

Advantages of Random Forests

As mentioned in the previous subsection, Random Forests are typically more robust to data changes and potentially incorporate more information than individual trees. A byproduct of this robustness is that random forests typically do not require an exhaustive search for the trees hyperparameters, as the ensembling algorithm compensate for the noise from the individual trees. In practice, mostly the only parameter considered when building random forests is the number of trees k. Typically, a larger k improves the prediction accuracy, at the expense of an increase in computational cost. This increase in accuracy naturally comes at the expense of lost interpretability, as the final decision is based on an average of multiple trees. Methods for interpretability are discussed in Chapter 8.

5.4 Boosted Trees

Boosted trees are an alternative ensemble learning technique that enhances the performance of decision trees by combining multiple *weak learners* (trees) to create a robust predictive model. The boosting algorithm aims to correct the errors of the previous models, sequentially improving the overall performance.[2]

The boosting process begins with the full training set. In the first step, the first tree is trained and the prediction errors are recorded. In the next step, the second tree is trained on the same data, but assigning different weights to those observations that were incorrectly predicted by the previous learner [3] The process of constructing trees and re-weighting the observations that were wrongly predicted is repeated M times, where M is a hyperparameter representing the number of trees in the model. Once all weak learners have been

[2]There are multiple algorithms that

[3]While the inclusion of the full training set and the use of weights are specific to the Adaptive Boosting (AdaBoost) and Gradient Boosting algorithms, these are the most usual algorithms used in practice.

trained, the final prediction is achieved as a weighted average in the case of regression problems, with the weights reflecting the performance of each tree:

$$\text{Final Prediction} = \sum_{i=1}^{N} \alpha_i h_i(\mathbf{x}) \tag{5.5}$$

where N is the number of weak learners, α_i is the weight of the i-th learner, and $h_i(\mathbf{x})$ is the prediction of the i-th learner.

The advantages and disadvantages of using boosted trees are generally similar to those of using random forests: while prediction accuracy is improved and non-linearity is captured, it comes at the expense of interpretability and added risk of overfitting. Computational costs are also a concern when dealing with large datasets and multiple iterations.

5.5 Support Vector Machine Regression

Support Vector Machines (SVMs) are versatile machine learning models initially designed for classification. *Support Vector Regression* (SVR), as suggested by its name, is a derivation that is particularly suited for regression tasks. In economic and financial applications, SVR is generally less popular than other machine learning algorithms such as neural networks and random forests. However, the model still provides the ability to capture non-linear relationships and is robust to outliers, reason why it has seen an increase in popularity among regression tasks and is included in this research.

Support Vector Regression is an extension of the SVM algorithm used for predicting continuous outcomes. Unlike traditional regression models that minimize the error between predicted and actual values, SVR aims to fit the best line (or hyperplane in higher dimensions) within a specified margin of tolerance. This margin, known as the epsilon (ϵ) margin, allows SVR to focus on capturing the overall trend of the data rather than individual data points, making it less sensitive to outliers.

The SVR model optimizes an objective function that balances two terms: minimizing the model's complexity (ensuring a flat hyperplane) and penalizing the errors of data points that lie outside the epsilon margin. The regularization parameter (C) controls this trade-off, where a higher C value puts more emphasis on minimizing errors, potentially leading to a more complex model.

Linear SVR For linear SVR, the objective function is:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) \tag{5.6}$$

subject to

$$\begin{aligned} y_i - (w \cdot x_i + b) &\leq \varepsilon + \zeta_i, \\ (w \cdot x_i + b) - y_i &\leq \varepsilon + \zeta_i^*, \\ \zeta_i, \zeta_i^* &\geq 0, \end{aligned}$$

where w is the weight vector, b is the bias, ζ_i and ζ_i^* are slack variables, and C is the regularization parameter.

Non-linear SVR The non-linear SVR objective function with a kernel is:

$$\min_{\alpha,\alpha^*} \frac{1}{2}(\alpha - \alpha^*)^T K(\alpha - \alpha^*) + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{5.7}$$

subject to

$$\begin{aligned} y_i - \sum_{j=1}^{N}(\alpha_j - \alpha_j^*)K(x_i, x_j) &\leq \varepsilon + \xi_i, \\ \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) &= 0, \\ \xi_i, \xi_i^* &\geq 0, \end{aligned}$$

where α and α^* are Lagrange multipliers, K is the kernel function, and ξ_i and ξ_i^* are slack variables.

The prediction for SVR is given by:

$$\hat{y} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x, x_i) + b \tag{5.8}$$

where N is the number of support vectors, and the formula simply represents the use of the hyperplane parameters.

Advantages and disadvantages of Support Vector Regression

Similarly to other machine learning algorithms, the main benefit of SVR lies in its ability to capture non-linear relationships, improving predictive accu-

racy, and its disadvantages in the lost of interpretability and computational complexity.

In addition, SVR are robust to outliers and their complexity is to some extent under the control of the modeler through the choice of hyperparameters. Along with this control over complexity, however, comes the challenge of finding the right set of hyperparameters.

5.6 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm that makes predictions based on the similarity of an observation to others used in the training process. It is a popular algorithm in economics and finance, mainly used for consumer behavior analysis, real estate valuation and credit risk assessment.

Given a training dataset with input features and target variable, the KNN algorithm for regression computes the distance between the instance and all observations in the training set using a distance metric (usually Euclidean distance), to determine which k observations are the most similar ones, the *k-nearest neighbors*. Then, the final prediction is calculated as the weighted average of the target value among the neighbors.

The K-Nearest Neighbors algorithm is a relatively simple and less computationally intensive than other algorithms like Neural Networks or Random Forests. KNN makes no assumptions about the underlying data distribution, allowing it to model complex relationships between economic variables without requiring predefined functional forms. While the connections are implicit in comparing instances based on the similarity of the features to produce the prediction, this algorithm does not calculate parameters for the features, nor does it compute operations on their values. Hence, model interpretability is reduced to purely simulated approaches.

Chapter 6

Data

The dataset used for this study ranges from January 2004 to December 2023 (240 months) and is comprised of 40 variables. The dependent variable or *target* is the EMAE, the official monthly GDP indicator, and the explanatory variables or *features* are a mix of financial variables, sectoral indicators, sentiment data, macroeconomic variables, and others. Both dependent and explanatory variables are described in more detail in the following subsections.

6.1 EMAE

The Monthly Economic Activity Estimator (EMAE) is an index published by the National Institute of Statistics and Censuses (INDEC) that provides a provisional estimate of the national Gross Domestic Product (GDP) on a monthly basis. Essentially, it involves a nowcasting exercise on GDP, which has a quarterly frequency and is presented with a lag of at least three months after the end of the reference quarter. The EMAE is published with a lag of between 50 and 60 days after the reference month, creating the need to observe alternative indicators for information on recent economic activity performance.

The indicator provides aggregated information for various sectors of economic activity:

1. Agriculture, livestock, forestry, and hunting
2. Fishing and related services
3. Exploitation of mines and quarries
4. Manufacturing industry

5. Electricity, gas, and water
6. Construction
7. Wholesale and retail trade and repairs
8. Hotels and restaurants
9. Transport, storage, and communications
10. Financial intermediation
11. Real estate, business, and rental activities
12. Public administration and defense
13. Education
14. Social and health services
15. Other community, social, and personal service activities
16. Services of private households employing domestic staff

6.2 Explanatory variables

The set of variables used in this exercise builds on a subset of the ones used in D'Amato et al. (2017), which consists of two groups of indicators: one with variables published with a delay of less than 10 days after the reference month, and another with variables published with a delay between 10 and 30 days. A total of 16 variables are taken from their study. This study incorporates 24 additional variables to take advantage of a broader range of information for enhanced GDP nowcasting. The inclusion of these variables stems from recent developments, either in the increased accessibility of certain data or in the capacity of advanced models to seamlessly integrate them. Table 6.1 shows the full set of indicators used as inputs in the model.

Since the original data is presented a different frequencies (financial and monetary variables are generally presented in daily frequency), all variables were aggregated to monthly frequencies. Additionally, the variables were seasonally and inflation-adjusted, as necessary. Lastly, as a pre-processing step, three sets of synthetic variables were included: lagged EMAE (from 1 to 12

lags), and two sets of running averages of all explanatory variables, using windows of 3 and 12 months.

Table 6.1: Set of Variables

	Variable	Additional Adjustments
1	Automobile production	
2	Automobile exports	
3	Vehicle sales to dealerships	
4	Cement dispatches	
5	Steel Production: Raw	
6	Steel Production: Iron	
7	Steel Production: Cold Rolled Steel	
8	Steel Production: Hot Rolled Steel	
9	Power Demand	
10	Income Tax Collection (Total)	
11	Income Tax Collection (DGI)	
12	Income Tax Collection (DGA)	
13	Value Added Tax Collection (Total)	
14	Value Added Tax Collection (DGI)	
15	Value Added Tax Collection (DGA)	
16	Other Activity-Related Tax Collection	
17	Other Trade-Related Tax Collection	
18	Other Import-Related Tax Collection	
19	M2	
20	Private M2	
21	Interest Rate	Adjusted by inflation expectations
22	S&P MERVAL Index	
23	Blue Chip Swap Spread	
24	Real Exchange Rate	Adjusted by BCS spread
25	Commodity Price Index	Adjusted by BCS spread
26	Deposits on Commercial Banks	
27	Loans Provided by Commercial Banks	
28	MoM CPI Inflation	
29	YoY CPI Inflation	
30	YoY Inflation Expectations	
31	FX Rate (US Dollar)	
32	Consumer Confidence Index: General	
33	Consumer Confidence Index: National	
34	Consumer Confidence Index: GBA	
35	Consumer Confidence Index: Personal Situation	
36	Consumer Confidence Index: Macroeconomic Situation	
37	Google Trends: Job	
38	Google Trends: Tourism	
39	Google Trends: Mercado Libre	
40	Google Trends: Wheels	

Chapter 7

Performance Evaluation

7.1 Error metric

To evaluate the performance of the models, the Absolute Error will be the primary metric. Defined as follows,

$$e_t = (\hat{y}_t - y_t)\,, \tag{7.1}$$

where $\hat{y}$ and y represent the estimate and actual value of the month-over-month percentage change of the GDP indicator, respectively:

$$y_t = \frac{\text{EMAE}_t}{\text{EMAE}_{t-1}} - 1$$

The error metric, e_t, then represents the forecast error in percentage points.

To calculate the time-series average error, we will use the *Mean Absolute Percentage Error* (MAPE):

$$MAPE = \frac{\sum_{t=1}^{n} |e_t|}{n},$$

where n is the number of *out-of-sample* observations.

7.2 Out-of-sample Testing

We employ a *walk-forward cross-validation* approach to assess the models' out-of-sample performance. This method involves iteratively training the models and individually predicting each data point in the test set, using all available information up to that point. Specifically, our test set spans 96 months, from

January 2016 to December 2023. For each month in this period, the model is trained on all prior data and then used to make a single prediction for that month. The steps in the testing process can be summarized as follows.

a. **Determine the initial training and evaluation sets**. In this exercise, the first 131 observations make up the train set, and the last 96 observations make up the test set. This implies a 60/40 split between the train and test sets.

b. **Evaluate the prediction of EMAE at time (t)**.

   1. Train the model with the information available up to time $(t-1)$ included.
   2. With the obtained parameters, estimate EMAE_t.
   3. Calculate the percentage deviation (e_t).

c. **Evaluate the prediction of EMAE at time ($t+1$)**.

   1. Retrain the model with the information available up to time (t) included.
   2. With the obtained parameters, estimate EMAE_{t+1}.
   3. Calculate the percentage deviation e_{t+1}.

d. **Repeat the process for each observation in the evaluation set and calculate** MAPE:

   - $\text{MAPE} = \frac{\sum_{t=1}^{96} e_t}{96}$

   This approach presents two benefits:

   - Ensures that the models are evaluated in a manner that closely mimics real-world forecasting scenarios, where future data points must be predicted based on historical information alone, avoiding the *look-ahead bias*. [1]

   - Captures the real-time benefit of re-training the model when new information becomes available, as parameters become less informative as the series evolve and relationships change.

[1] Banbura, et. al. (2013) presents the distinction between real-time and *pseudo real-time* out-of-sample forecast evaluation based on the presence of data vintages. In most studies, when data vintages are not available, final data is used, ignoring the potentially significant impact of revisions in the data. This naturally biases the performance evaluation by not fully reflecting the information available at the time of a real-life use of the models.

7.3 Results

Table 7.1 presents the out-of-sample performance of various machine learning models (columns) across different filtering techniques (rows) as measured by the Mean Absolute Percentage Error (MAPE), and the standard deviations of the errors between under them. To provide additional context to the error metric, the performance of the models is compared to that of an AR(1) model where the only independent variable is the lagged target variable by one period.

The results indicate that, with the exception of the Neural Network, all models outperform the AR(1) benchmark in terms of MAPE, irrespective of the feature selection technique used. This finding is consistent with the existing literature on machine learning algorithms, although it contrasts with previous research that typically identifies Neural Networks as the best-performing algorithm. If the mean error of each algorithm is relativized to its standard deviation, however, the Neural Network and the KNN algorithm show a decrease in performance against the benchmark AR(1).

An additional finding is the moderte impact of the filtering technique on model performance. Most models achieve their lowest MAPE when the data is filtered using the ANOVA F-test (*ANOVA*). Filtering based on Mutual Information (*MI*) results in a slightly higher MAPE in most algorithms, and the combination of both sets of features leads to a significantly higher MAPE. These results suggest that models perform better with fewer features, as filtering based on ANOVA leads to a smaller set of variables. This is possibly due to low signal-to-noise ratio in some variables; removing these variables may lead to more accurate learning.

Lastly, when the exercise is run on data without any filtering, the resulting MAPE is at least as high as with the filtered dataframes.

7.3.1 Statistical significance of the results

As a final step, a Diebold-Mariano Test is conducted to evaluate the statistical significance of the performance differences among the models. Results for this exercise are shown in Tables 7.2 to 7.2.

We start by comparing each model's performance against the AR(1) benchmark. The results indicate that the Gradient Boosting model consistently outperforms the AR(1) model across all specifications, with differences found to be statistically significant at the 10% level. Furthermore, the improvement

Table 7.1: Out-of-sample Performance

	NN	**RF**	**SVR**	**GB**	**KNN**	**AR(1)**
Full dataset	0.021	0.015	0.017	0.014	0.018	0.019
	(0.020)	(0.020)	(0.020)	(0.017)	(0.019)	(0.022)
ANOVA	0.023	0.014	0.016	0.014	0.016	0.019
	(0.022)	(0.016)	(0.020)	(0.017)	(0.017)	(0.022)
MI	0.021	0.015	0.017	0.014	0.017	0.019
	(0.019)	(0.020)	(0.020)	(0.017)	(0.017)	(0.022)
ANOVA + MI	0.019	0.014	0.017	0.014	0.017	0.019
	(0.019)	(0.019)	(0.020)	(0.018)	(0.018)	(0.022)

of the Random Forest model over the benchmark is confirmed only under the ANOVA filtering specification. For the other models, we are unable to reject the null hypothesis, suggesting that their performance does not significantly differ from that of the AR(1) model.

Additionally, we compare all model's performance against each other for each specification. The results show that only under certain specifications the models are different from each other at a statistically significant level.

When adjusting our results for statistical significance, this study can only support existing literature suggesting that machine learning algorithms outperform a benchmark to some extent. Several factors may contribute to this outcome: the data not being fully processed (as we intentionally refrained from applying PCA or autoencoders to maintain interpretability), the available indicators for Argentina exhibiting a low signal-to-noise ratio, or the models not being calibrated to their full potential.

It is important to note that Argentina's economy over the past 30 years has been characterized by abrupt changes in economic regimes and shifts in the country's economic structure. While machine learning algorithms are generally expected to capture complex relationships more effectively than traditional models, the constantly evolving nature of the underlying economic system may hinder their ability to learn from the complete dataset. In this context, additional steps can be taken to improve the predictive accuracy, such as using a moving window for training, or incorporating other variables such as mobility, traffic, payments, and sentiment data. The application of these suggestions is left for future studies.

Table 7.2: Diebold-Mariano Test - Full Dataset

	NN	**RF**	**SVR**	**GB**	**KNN**	**AR(1)**
NN		0.193	0.082	0.005	0.012	0.901
RF			0.656	0.233	0.840	0.316
SVR				0.098	0.392	0.500
GB					0.103	0.089
KNN						0.386
AR(1)						

Table 7.3: Diebold-Mariano Test - ANOVA

	NN	**RF**	**SVR**	**GB**	**KNN**	**AR(1)**
NN		0.001	0	0.002	0	0.433
RF			0.104	0.352	0.218	0.078
SVR				0.213	0.155	0.397
GB					0.65	0.091
KNN						0.159
AR(1)						

Table 7.4: Diebold-Mariano Test - Mutual Information (MI)

	NN	**RF**	**SVR**	**GB**	**KNN**	**AR(1)**
NN		0.357	0.208	0.085	0.016	0.924
RF			0.601	0.228	0.965	0.259
SVR				0.178	0.265	0.468
GB					0.387	0.09
KNN						0.285
AR(1)						

Table 7.5: Diebold-Mariano Test - ANOVA + MI

	NN	**RF**	**SVR**	**GB**	**KNN**	**AR(1)**
NN		0.297	0.664	0.102	0.012	0.632
RF			0.460	0.276	0.833	0.210
SVR				0.241	0.188	0.496
GB					0.495	0.095
KNN						0.280
AR(1)						

Chapter 8

Interpretability

As previous studies have shown, and this study supports partially, machine learning algorithms have the potential of providing more accurate estimates of GDP than traditional econometric techniques, and hence improving decision making and potentially our overall understanding of the dynamics driving GDP. In order to grasp these models' full potential, however, the challenge of interpretability has to be overcome.

As outlined in Chapter 3, the main and most common disadvantage of machine learning algorithms is their black-box nature. And it is this challenge that halts their adoption in practice: understanding the model being used is as important as the final estimation and, often times, even more. The interest in interpretability is different depending on the goal of the researcher: while academic research may be mainly interested in understanding the underlying relationships between variables, forecasters and analysts may be more interested in understanding how a certain estimate is reached, identifying its main drivers and potential errors, and policymakers will be interested in both.

In this chapter, the main aspects to the issue of *eXplainable AI*, popularly referred to as *XAI*, are briefly introduced and a number of interpretability methods that can be used in the context of macroeconomic nowcasting and forecasting are exposed. The aim of this chapter is then to assess whether this challenge can be sorted and if the questions typically asked by researchers can be answered through these models.

As a comprehensive discussion of the topic of interpretability falls beyond the scope of this paper, for a more in-depth exploration of these methods, the work in Molnar (2024) is highly recommended. There is much written on machine learning interpretability, and the focus of this study is not to propose

new methods, but rather to bring them into the context of economic modeling.

8.1 Types of interpretability

Generally, interpretability methods can be classified according to three main criteria: intrinsic vs. post hoc methods, model specificity vs. model agnosticity, and global vs. local interpretability.

8.1.1 Intrinsic and Post Hoc Methods

Intrinsic interpretability refers to models that possess a structure that is inherently understandable. The *par excellence* example of intrinsic interpretability is that of a linear regression, as the impact each variable has on the output is fully available in the coefficients. A short regression tree is an additional example of intrinsic interpretability, as understanding the rules over a handful of variables is typically within reach for the modeler, specially if the variables have a simple interpretation (i.e., are not composites of other variables). **Post hoc interpretability**, conversely, involves applying interpretive methods to the model after it has been trained. They provide insights into model behavior by analyzing the model's behavior and predictions outside of the training process. Permutation importance is a popular example of post hoc interpretability techniques; this method quantifies the contribution of each feature by measuring the degradation in model performance when the values of that feature are randomly shuffled. A more detailed description of this and other techniques is provided in the following subsection.

While intrinsic interpretability is generally a desirable characteristic, it comes tied to simple models, which often present a cost in terms of accuracy. For that reason, as practical applications steer towards more complex models to gain prediction accuracy, intrinsic interpretability is increasingly reduced, and the need for post hoc interpretability methods grows.

8.1.2 Model Specific and Model Agnostic Methods

Model-specific methods are tailored for particular types of models, delivering clear interpretations relevant to those models. The coefficients of a linear regression or feature importance metrics in tree-based models are prime examples of model-specific interpretability, as they are derived specifically from

the models' structures. On the other hand, **model-agnostic** methods are those that can be utilized with any machine learning model, regardless of their structure. These methods involve techniques that are general enough to be applied across different models. Permutation Importance is also an example of a model-agnostic interpretability method.

While the examples provided for model-specific methods happen to be intrinsic, this needs not be the case; a post-hoc example of model-specific techniques is Layer-wise Relevance Propagation (LRP), a method designed specifically for interpreting Neural Networks. The opposite is, however, not true: all model-agnostic methods are post hoc in nature.

8.1.3 Local and Global Methods

Local interpretability methods focus on individual predictions, offering explanations for specific outcomes or for the model through the lens of a particular observation. The LRP technique mentioned above is an example of a local method, as it traces back the generation of a particular prediction to get a measure of the importance each variable had in it. **Global interpretability** methods, conversely, aim to explain the overall behavior of the model. These explanations are not tied to specific predictions, but rather to the entire dataset. Feature importance are usually measured using global interpretability techniques (although not always).

8.2 Interpretability Methods

Interpretability in machine learning refers to the degree to which the internal workings and predictions of a model can be understood by humans. Additionally, when used for economic modeling, interpretability should be expanded beyond the understanding of the mechanics of the algorithms, to the understanding of the underlying relationships between variables and the general dynamics of the system trying to be captured by the models. When using machine learning algorithms for economic estimations, researchers typically seek to understand aspects such as how different variables influence the model (and system), which variables are deemed most significant, how a specific prediction is achieved, etc. The choice of interpretability method can vary based on the question to be answered and the specific models used. In this subsection, a number of techniques are explained and tested.

8.2.1 Feature Importance

One of the most popular questions in economics is what are the most important drivers of a variable of interest. Researchers typically attempt to quantify the impact that one explanatory variable has on the response variable to understand how they are related and potentially how that relationship can be used. Naturally, this question is answered through the use of economic models, and there are multiple techniques through which this information can be retrieved from machine learning algorithms.

Feature importance techniques can be both intrinsic and post hoc, model-specific and model-agnostic, local and global. Intrinsic measures of feature importance can be found in tree-based algorithms like Random Forests and Gradient Boosting, as they are calculated directly from the informational gain in the training process. Alternatively, Layer-wise Relevance Propagation is a model-specific, post hoc measure of feature importance for Neural Networks. There are model-agnostic alternatives, such as permutation importance or SHAP values.

Additionally, feature importance can be interpreted both as an ordinal or a cardinal measure, depending on the interest of the model. In this section, both views are considered.

Feature importance in tree-based algorithms

As part of their training process, regression trees measure the *informational gain* (IG) derived from splitting the data based on every possible variable, and performs the split on the variable that maximizes it. This metric constitutes an intrinsic measure for feature importance in regression trees.

Naturally, this metric can also be computed for ensembles of trees such as Random Forests and Gradient Boosting, by aggregating the IG of each feature across the different trees that make the ensemble model. Particularly, a feature importance metric for these algorithms can be the simple average of the IGs associated to each feature:

$$FI_i = \frac{1}{n} \sum_{1}^{n} IG_{i,n}$$

where i indicates the feature, and n the number of trees in the ensemble model.

Table 8.1 displays the ten most important features identified by the Random Forest model, along with their associated importance scores. It also presents

the importance metrics for the same set of features using Gradient Boosting. Each model includes two columns: the first column indicates the ranking position of the feature based on importance, while the second column presents the corresponding feature importance score.

A notable observation from the table is the overall similarity between the two rankings. Both models assess the importance of features in a comparable manner, with cement dispatches and vehicle production occupying the top two positions in both rankings. Furthermore, seven out of the ten features are common to both models' top rankings, although their positions differ. This finding is consistent with the comparable learning characteristics of the two models. It should be noted that this metric provides a measure of how the models' accuracy generally improves based on a feature, but it does not provide information on the direction or shape of the feature's impact.

Table 8.1: Feature Importance - Tree-Based Algorithms

Feature	**RF (1)**	**RF (2)**	**GB (1)**	**GB (2)**
cement	1	0.190	1	0.240
vehicles_production	2	0.117	2	0.143
import_taxes	3	0.066	5	0.073
cement_ma_3	4	0.044	13	0.012
raw_steel_sa	5	0.039	21	0.008
iva_dga	6	0.037	14	0.011
vehicles_exports	7	0.034	4	0.074
m2_prive_ma_3	8	0.032	9	0.019
cement_ma_12	9	0.028	3	0.101
iva	10	0.027	6	0.044

Layer-wise Relevance Propagation

Another model-specific method is the Layer-wise Relevance Propagation (LRP). LRP is a post hoc interpretability method specifically designed for deep learning models, particularly Neural Networks.

The fundamental idea behind LRP is to attribute the prediction made by the model back to the input features by propagating the relevance scores from the output layer to the input layer. This is accomplished through a set of rules that distribute the prediction score (or relevance) of a neuron back to its input features based on their contribution to the neuron's activation. The process starts with the output layer, where the final prediction is assigned a relevance

Table 8.2: Layer-wise Relevance Propagation - Global

	Rank (Train)	Relevance Score (Train)	Rank (Test)	Relevance Score (Test)
dollar_off	1	0.522	1	0.675
iva_dga_ma_3	2	-0.496	2	-0.71
hot_lam_sa_ma_3	3	-0.312	6	-0.437
gtrends_wheels_ma_3	4	0.329	3	0.478
cold_lam_sa_ma_12	5	-0.273	5	-0.402
power_demand	6	0.313	8	0.43
gap_blue_ma_3	7	-0.294	4	-0.448
inflation_mom_ma_3	8	-0.242	30	-0.278
ganancias_dgi_ma_12	9	0.249	11	0.406
ganancias_ma_12	10	0.227	9	0.37

score, which is then propagated backward through the network layers. Each neuron in the hidden layers receives a portion of the relevance score based on its contribution to the activations of subsequent neurons.

While Layer-wise Relevance Propagation (LRP) is inherently a local method, it can also be utilized to gain insights into the overall model by calculating relevance scores across a large set of instances and then averaging these scores for each variable. Table 8.2 displays the normalized average relevance scores of the ten most important features across both the training and test sets. One notable finding is that the relevance assessments for both sets are very similar, indicating that averaging across multiple instances effectively captures the overall mechanics of the model.

Regarding the selected features, the differences compared to the feature importance rankings derived from tree-based models are significant. None of the variables identified as most important in the Random Forest and Gradient Boosting models appear in the Neural Network's list of key features. This highlights how different models can interpret and utilize the same data in distinct ways. This pattern will be further explored throughout this section.

LRP for local interpretability

By employing Layer-wise Relevance Propagation (LRP), researchers can measure the contribution of each variable to the final prediction, thereby gaining insight into the model's decision-making process. Table 8.3 presents the top 10 normalized relevance scores for the May 2022 prediction made by the Neural Network model. The table reveals that several features have relatively similar relevance scores in absolute values, suggesting that the prediction is influenced not by a single variable, but rather by the interaction of multiple variables.

Similarly to the intrinsic feature importance from tree-based algorithms, LRP's specificity to Neural Networks poses a limitation as it cannot be used

Table 8.3: Layer-wise Relevance Propagation - Local

Feature	Relevance Score
dollar_off	0.430
iva_dga_ma_3	-0.396
gtrends_wheels_ma_3	0.375
power_demand	0.313
steel_sa_ma_3	0.291
hot_lam_sa_ma_3	-0.289
raw_steel_sa_ma_12	0.273
cold_lam_sa_ma_12	-0.265
emae12	-0.257
gap_blue_ma_3	-0.235

to explain other algorithims. Researchers utilizing other models must seek alternative methods for feature importance analysis. Therefore, in the following sections, we will explore model-agnostic approaches to address this limitation, namely *permutation importance and SHAP*.

Permutation Importance

A popular post hoc model-agnostic method for measuring feature importance is *permutation importance*. This technique involves evaluating how the performance of a trained model degrades when the values of a specific feature are randomly shuffled or permuted. By disrupting the relationship between the feature and the target variable, the method simulates the presence of a non-informative feature. By quantifying the change in accuracy, researchers can infer the contribution of that feature to the overall performance of the model. A greater decrease in accuracy indicates a higher importance of the variable, while a smaller drop suggests that the feature has little relevance to the predictions.

The methodology for feature permutation typically involves several steps. First, a baseline performance metric (e.g., accuracy, F1 score, or mean squared error) is established using the original dataset. Subsequently, the values of the feature being analyzed are randomly shuffled, and the model's performance is reassessed on this permuted dataset. This process is repeated multiple times to ensure robustness and to account for variability. The final importance score for the feature is calculated as the average decrease in the model's performance across all permutations.

One of the primary advantages of permutation importance is its model-

agnostic nature, which allows researchers to compute a comparable metric across different models. The results of calculating permutation importance across all models in this study are presented in Table 8.4. For this table, the most important features for each model are selected, provided that they are statistically significant. This significance is determined by ensuring that the mean importance of the feature across all permutations is greater than two times its standard deviation.

Table 8.4 illustrates the variations in the way each algorithm learned from the data. Although certain features are recognized as important across all models (again cement dispatches and vehicle production), there are significant differences regarding the number of features deemed important and the specific variables identified. Features that rank among the most important in one model may not be considered significant in another, highlighting the diverse influences the variables have depending on the algorithm being used.

As in the case of the intrinsic feature importance measure in tree-based algorithms, permutation importance also does not provide information on the direction or shape of the impact of the variables in the models. Additionally, it is important to note that the permutation importance metric does not distinguish between individual effects and interaction effects when assessing the impact of a variable. If a variable influences the model through both an individual channel and through an interaction with another variable, the measured decrease in model performance will reflect both influences without indicating their specific contributions. Lastly, the presence of correlated variable presents to potential issues: first, it can lead to bias in the results due to unrealistic simulated permutations; second, the importance of correlated features may be underestimated as the models can learn from one instead of the other, hence making each individual variable less important. A possible workaround this problem is to combine the correlated features' importance, although additivity is not necessarily a characteristic of the metric.

Table 8.4: Permutation Importance

Rank	NN	RF	GB	SVR	KNN
1	cement	vehicles_production	vehicles_production	cement	gtrends_tourism_ma_12
2	deposits_ma_12	cement	cement	power_demand	IPMP_ccl
3	vehicles_production	ganancias_dga_ma_3	-	emae12	iva_dgi_ma_12
4	-	loans_ma_3	-	hot_lam_sa_ma_12	vehicles_sales_ma_3
5	-	vehicles_exports_ma_3	-	-	gtrends_wheels_ma_3
6	-	merval_ma_12	-	-	emae12
7	-	payroll_taxes_ma_3	-	-	ganancias_dga_ma_3
8	-	raw_steel_ma_12	-	-	iva_ma_3
9	-	-	-	-	loans_ma_3
10	-	-	-	-	iva_dgi

SHAP

SHapley Additive exPlanations, commonly referred to as *SHAP* is another post hoc, local model-agnostic framework for interpreting the predictions of machine learning models. Grounded in cooperative game theory, the method was first proposed by Lundberg and Lee (2017). In essence, Shapley values measure the contribution of each feature to the overall prediction by computing how the predictions change when the value of the feature varies *over all possible combinations of the other features.*[1] The final contribution is then the average contribution of the feature across all the computed combinations.

One of the primary advantages of SHAP is its ability to produce additive explanations, meaning that the prediction of a model can be expressed as the sum of the contributions from each feature. This property facilitates interpretability, as stakeholders can easily understand the impact of individual features on model predictions —a particularly useful application in economics, where accountability of predictions is a paramount requirement. SHAP provides the contribution of each feature to the prediction on top of the base value, which is the average estimated value of the target variable. Moreover, as in the case of permutation importance, it also can be applied to a wide range of machine learning algorithms. Its versatility has led to widespread adoption in various fields, including finance.

While SHAP is primarily a local method, it can be utilized to assess feature importance in the overall model by applying the technique across multiple observations and averaging the SHAP values for each feature. Table 8.5 presents the feature importances as calculated through SHAP values for all models in this study. Each model is represented with two columns: the first column indicates the ranking of the feature in terms of importance, while the second column displays the *median* SHAP value of the features, expressed in percentage points.

The results shown in Table 8.5 align qualitatively with those obtained through permutation importance to some extent. Several variables consistently rank among the most important features across all models, yet the differences in rankings further illustrate how each model processes the data in different ways. A particularly notable example is the Neural Network model, which identifies only *iva_dga* as one of its most significant features.

[1] Testing all possible combinations can get computationally expensive quickly as datasets grow larger. In reality, most practical implementations of SHAP only compute a subset of the possible combinations.

These differences in rankings are expected due to the inherent differences in how they are calculated. Specifically, the intrinsic measure of feature importance used in tree-based algorithms assesses a variable's ability to accurately split the target variable, but it does not account for the size of that impact. In contrast, SHAP values reflect the size of the effect. In addition, tree-based feature importance measures the feature's impact individually, hence not reflecting possible interaction effects with other variables, while SHAP accounts for these interactions.

The main addition in this analysis is the inclusion of the median SHAP values, which represent the typical impact of each variable on the predictions. For instance, the *cement* variable, according to the Random Forest model, typically increases the prediction by 0.145 percentage points. Alternatively, the *vehicles_production* variable typically decreases the prediction by 0.092 percentage points. This measure, however, should not be interpreted as reflecting a negative relation between *vehicles_production* and the target variable, but rather the typical impact conditional on the feature's distribution. If a feature that correlates positively with the target variable generally takes negative values, it will typically have a negative SHAP value. This is the case of the *vehicles_production* variable, as will be shown in Figure 8.1.

Table 8.5: SHAP-based Feature Importance

Feature	RF (1)	RF (2)	NN (1)	NN (2)	GB (1)	GB (2)	SVR (1)	SVR (2)	KNN (1)	KNN (2)
cement	1	0.145	42	0.000	1	0.128	2	0.000	4	0.000
vehicles_production	2	-0.092	95	0.000	2	0.001	97	0.000	95	-0.001
import_taxes	3	-0.094	19	0.006	4	0.008	4	-0.027	3	-0.027
iva_dga	4	-0.088	9	0.008	14	-0.039	5	-0.022	5	-0.031
iva	5	-0.082	65	0.000	3	-0.136	6	-0.011	6	-0.024
cement_ma_3	6	0.005	67	0.001	10	-0.004	9	-0.016	13	0.000
ganancias_dga	7	-0.066	41	-0.004	7	-0.045	10	-0.021	11	-0.017
trade_taxes_ma_3	8	-0.036	84	0.000	39	-0.001	31	-0.007	7	-0.018
gtrends_tourism_ma_12	9	-0.005	71	0.000	12	-0.025	16	0.000	8	0.000
icc_gba_ma_3	10	-0.023	75	0.000	26	-0.013	38	0.000	23	0.000

An alternative visualization technique for interpreting SHAP values is the *summary plot*. Figure 8.1 presents a summary plot for the Random Forest model, created using the full test set.[2] This plot displays the most important features ranked by their average SHAP values on the y-axis, with the importance decreasing from top to bottom. The x-axis represents the SHAP values, which indicate the magnitude and direction of the feature's impact on the model's predictions for each instance.

In this visualization, each point corresponds to a specific observation in the test set, illustrating the contribution of the feature to that instance's predic-

[2]All SHAP-related plots were created using Pythons's *shap* library.

tion. The color gradient of the points indicates the feature values, with red representing higher values and blue indicating lower values. By examining the distribution of these points, we can gain insights into the typical effect size and direction of each feature across the dataset, allowing for a clearer understanding of how each feature influences the model's predictions.

Figure 8.1 further illustrates the observation that features positively correlated with the target can exhibit negative SHAP values. In the second row, the distribution of SHAP values for the *vehicles_production* variable is displayed. The positive correlation is evident, as lower values of this feature correspond to negative contributions to the predictions, while higher values are linked to positive contributions. The typical negative contribution is then primarily attributed to the feature's negatively skewed distribution, which leads to negative impacts occurring more frequently than positive ones.

SHAP for local interpretability

Having established the overall feature importances derived from SHAP values, it is essential to recognize that SHAP is particularly valuable for local interpretability as well. This capability allows users to understand the contribution of individual features to specific predictions, providing insights into the model's behavior at the instance level.

Table 8.6 illustrates the contribution of each feature to the February 2022 prediction according to the Random Forest model. The SHAP values indicate how the final prediction of -0.01 was reached: building on the base value of the target variable, 0.001, each variable's contribution to the final prediction is the SHAP value. For example, the *import_taxes* variable decreased the prediction by 0.313 percentage points, while the *cement* variable increased it by 0.166 percentage points, partially offsetting the negative impact. Overall, the cumulative contributions of all features explain the difference between the predicted value and the average value of the target variable.

The contributions to a specific prediction can also be represented visually in a clear and effective manner. Figures 8.2 and 8.3 illustrate the March 2022 predictions for the Random Forest and Gradient Boosting models, respectively. These figures display the final prediction as the sum of the contributions from each feature, along with the *base value*.[3]

[3]While "counteracting forces" conveys the idea of opposing contributions, it may not be the best choice of words in this context, as SHAP terminology describes features as players that work together to achieve the final outcome.

Figure 8.1: SHAP - Summary Plot

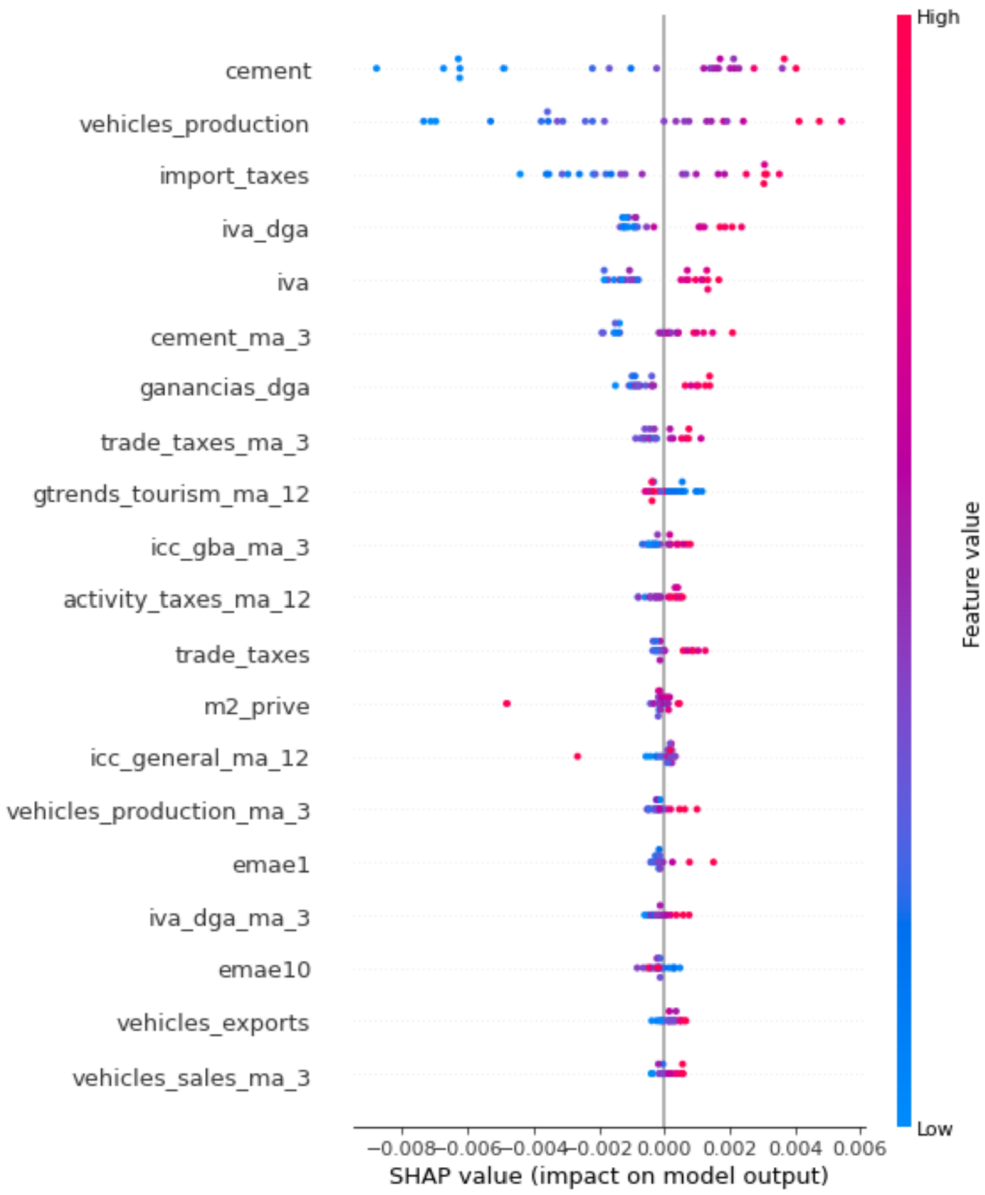


Figure 8.2: SHAP Values - Local - Random Forest

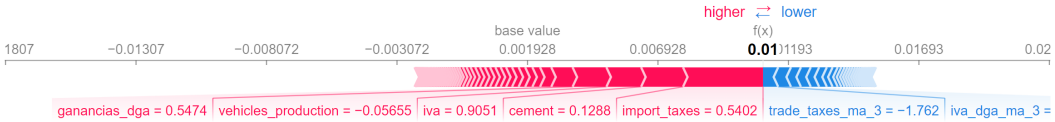


Figure 8.3: SHAP Values - Local - Gradient Boosting

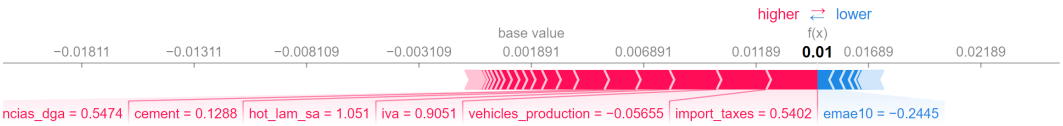


Table 8.6: SHAP Contributions - Random Forest

Feature	SHAP Value (p.p.)
import_taxes	-0.313
cement	0.166
iva	-0.154
raw_steel_sa	0.129
cement_ma_3	0.099
trade_taxes_ma_3	-0.088
iva_dga	-0.087
ganancias_dga	-0.084
m2_prive_ma_12	0.062
others	0.259
Total	**-0.011**

8.2.2 Shape of the Relationships

While the focus of the chapter so far has been put on methods for measuring feature importance in both local and global scopes, this section addresses methods that help explain *how* the variables affect the model and its prediction; i.e., what is the functional shape of the relationship between the variables and the target. For this, 2 popular model-agnostic methods are considered: Partial Dependence Plots (PDPs) and Surrogate Models.

Partial Dependence Plots (PDPs)

A Partial Dependence Plot (*PDP*) is a visualization tool used to illustrate the relationship between one or more features and the target variable. The plot displays the predicted values of the target variable on the y-axis against various levels of the feature on the x-axis. To create a PDP, the model is evaluated across a range of potential feature values while keeping all other features constant. PDPs can be viewed as a post hoc, model-agnostic alternative to the coefficients of linear regression. In a linear model, coefficients indicate that the target variable will increase by β with every unit change in the explanatory variable, regardless of the variable's range. However, with more complex machine learning models, the relationships between features and the target are often non-linear, making it non feasible to accurately represent these effects with a single coefficient. PDPs enable the assessment of a feature's marginal contribution to the model's overall predictions across different value ranges, providing a more nuanced understanding of these relationships.

Figure 8.4: Partial Dependence Plot - Random Forest

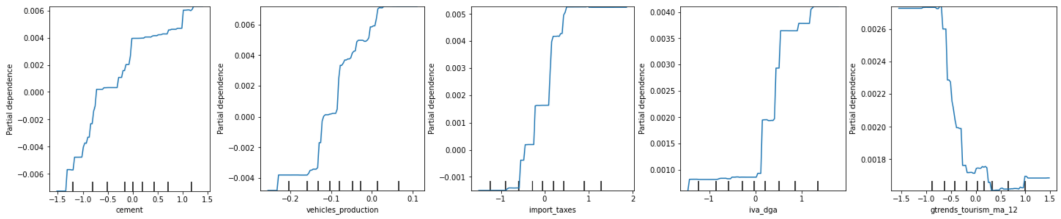


Figure 8.4 presents Partial Dependence Plots (PDPs) for a selection of variables using the Random Forest model. From this Figure, there are three main takeaways. First is the positive influence of the *cement*, *vehicles_production*, *import_taxes*, and *iva_dga* variables on the target variable, and the negative effect of the *gtrends_tourism_ma_12* variable. Second is the non-linear nature of the relationships. For instance, while *cement* positively affects the target variable, the magnitude of its impact diminishes as its value increases. Conversely, *iva_dga* initially has a minimal effect on the target in the lower half of its range, but its influence grows significantly in higher ranges. The third comes from observing the wider range of values in the y-axis scale for the first variables, which indicates that their impact is quantitatively more meaningful than the last variables. This aligns with the findings from the feature importance assessments, where we found that *cement* and *vehicles_production* were consistently the most impactful features according to the model.

An additional observation is the pronounced jaggedness of the Random Forest plots. This characteristic stems from the nature of tree-based algorithms' learning process, where the models build on decision trees that split the data at discrete thresholds. As a result, the impact of each variable is captured in a piecewise manner, leading to abrupt changes in the predicted values at specific levels of the features.

The same analysis is conducted for the KNN and Support Vector Regression models, with their results presented in Figures 8.5 and 8.6, respectively. Comparing these figures to Figure 8.4 reveals significant differences in how each model interprets the impact of the variables. While all models agree on the direction of the impact, their functional forms differ considerably.

First, the jaggedness observed in the Random Forest plot is absent in the KNN and SVR models, which exhibit smoother relationships. Second, the degree of non-linearity varies across the models; KNN offers a more gradual representation of the relationships than the Random Forest, whereas the SVR algorithm tends to interpret these relationships as nearly linear. This variation

Figure 8.5: Partial Dependence Plot - K-Nearest Neighbors

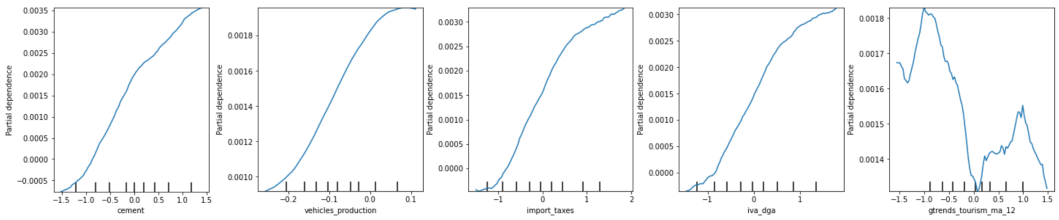
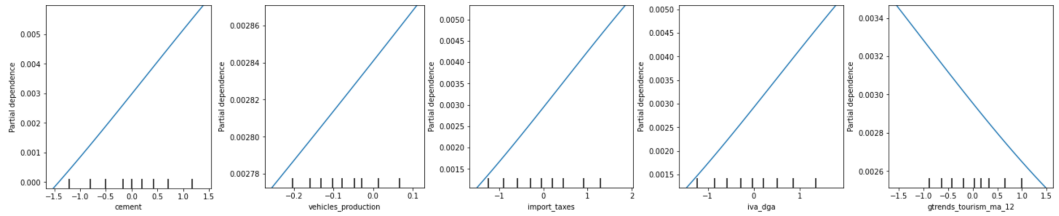


Figure 8.6: Partial Dependence Plot - Support Vector Regression



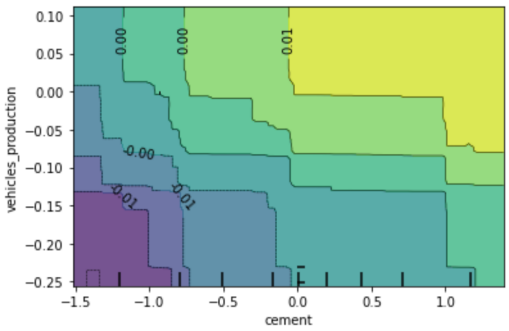
underscores the influence of the model choice on understanding relationships, highlighting a potential limitation of using machine learning algorithms in economics. This topic will be further explored in the conclusion of this chapter.

Bivariate PDPs

Partial Dependence Plots are a widely used interpretability technique due to their straightforward and intuitive explanations. A notable application of this technique is presented by Kohlscheen (2021), who utilizes PDPs to demonstrate how certain variables influence forecasted inflation according to a regression tree algorithm. However, these plots—particularly those involving a single variable—have two key limitations. First, they illustrate only the *marginal* impact of a variable without considering interactions between multiple variables. Second, they do not account for feature correlation. In datasets containing highly correlated features, which is often the case in economics, altering the value of one variable while holding others constant can produce a set of simulations that does not accurately reflect the data distribution and becomes unrepresentative. Although PDPs effectively convey the model's inner mechanics, in this context they will fall short in representing the underlying relationships among variables.

To address these limitations, researchers can employ several techniques. To account for interactions between variables, bivariate PDPs can be utilized, which illustrate the response of the target variable to different values of two variables simultaneously. To tackle the issue of representativeness due to fea-

Figure 8.7: Bivariate Partial Dependence Plot - Random Forest



ture correlations, Accumulated Local Effects (ALE) plots can be implemented, as they use the conditional distribution rather than the marginal distribution. An example of a bivariate PDP will be shown below.

Figure 8.7 illustrates a bivariate Partial Dependence Plot (PDP) depicting the marginal effects of the *cement* and *vehicles_production* variables, as well as their interaction, for the Random Forest model. This plot reinforces the findings from the univariate PDPs, indicating that both features positively influence the target variable.

The key advantage of this bivariate plot is that it reveals how the impact of one variable varies depending on the value of the other. Specifically, at lower levels of the *cement* variable, the effect of *vehicles_production* appears to be more pronounced compared to its impact at higher levels. Overall, the plot qualitatively suggests that both variables synergistically contribute to increasing the target variable. While this particular study does not find any pair of variables for which the interaction effect would diminish their individual contributions, bivariate PDPs are well-equipped to identify such relationships if they exist.

Although bivariate PDPs enhance understanding of interaction effects between two variables, they are limited to examining only two variables at a time, as visualizing higher-dimensional interactions is not feasible. Therefore, it is crucial to carefully select the variables included in the plot based on an initial assessment of the interaction effects among them.

Visual alternatives to PDPs

The disadvantages of PDPs, such as being vulnerable to correlation in the features, their inability to capture heterogeneous effects that do not depend on the feature's value, can be overcome with the use of alternative methods such

as Accumulated Local Effects plots (ALE) and Individual Conditional Expectation curves (ICE). The former provide a workaround to the independence assumption of PDPs by using the conditional distribution of the feature instead of the marginal one; the latter are able to capture heterogeneous effects by testing the impact of each variable on individual instances and showing the resulting predictions on a granular level.

Surrogate Models

Surrogate models are simplified, interpretable models that approximate the behavior of more complex and opaque machine learning models, such as ensemble methods or deep Neural Networks. These surrogate models are typically easier to understand and can help understand the underlying decision-making processes of the original models.

The value of surrogate models lies in their capacity to bridge the gap between model accuracy and interpretability. They are especially useful when the original model is too complex for stakeholders to understand or when regulatory requirements demand clear explanations. One common approach to creating a surrogate model involves fitting a linear regression to the predictions of a more complex model, using the same input features.

A surrogate model was developed based on the predictions of the Support Vector Regression (SVR) model in this study, with the most important variables and their coefficients presented in Table 8.7. However, the results reveal some unintuitive relationships, indicating that interpretations derived from this method may be misleading. First, let us interpret the table before assessing the results further. The coefficients are straightforward to interpret: according to the surrogate model, a unit increase in the *ganancias_dgi_ma_12* variable is associated with an average increase in the prediction by the SVR model of 2.554%, a unit increase in the *ganancias_ma* variable leads to an average decrease of 1.984%, and so on.

These results show two issues. First, the assessment of variable importance differs significantly from that in the original model. Second, unintuitive coefficients; when using a linear regression model as a surrogate, correlation between features can pose challenges. The coefficients of correlated features may become misleading, as they do not accurately represent each feature's unique contribution to the prediction. This undermines the interpretability of the surrogate model. For instance, the opposing effects of the features *ganan-*

Table 8.7: Feature Coefficients - Surrogate Model on the SVR

Variable	Coefficient
ganancias_dgi_ma_12	2.554
ganancias_ma_12	-1.984
activity_taxes_ma_12	-0.800
iva_dga_ma_3	0.729
import_taxes	0.524
import_taxes_ma_3	-0.466
m2_prive_ma_12	-0.350
iva_dga	-0.329
dollar_off	-0.317
icc_gba_ma_12	-0.313

cias_dgi_ma_12 and *ganancias_ma_12*, which should capture essentially the same dynamics, serve as evidence of this issue.[4]

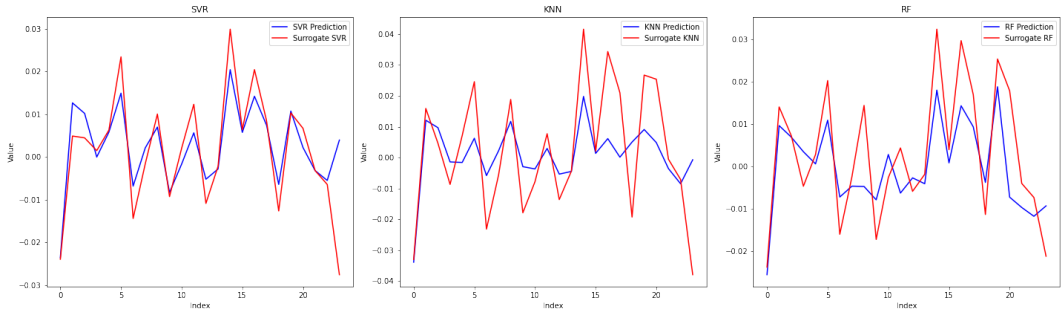
Furthermore, surrogate models are limited in their applicability and can only be employed under specific conditions. Since the primary goal of a surrogate model is to explain the inner mechanics of the original algorithm, it is essential that the simpler model effectively predicts the outputs of the more complex model. In addition to the exercise conducted for the Support Vector Regression (SVR) model, a linear model was also fitted on the predictions of the Random Forest and K-Nearest Neighbors (KNN) models. Their out-of-sample accuracy is illustrated in Figure 8.8. The poorer performance of the surrogate models when applied to the Random Forest and KNN models, compared to the SVR, indicates that these models are unlikely to provide an accurate representation of the underlying mechanics of the original algorithms.

8.3 Conclusion on interpretability

In testing various interpretability methods, it becomes clear that different models comprehend the underlying relationships between variables in distinct ways. Each technique assigns varying degrees of importance to individual features and interprets functional forms with subtle differences. While these variations are not overly pronounced, they raise questions about the adequacy of these models

[4]A popular alternative to linear regression as the surrogate model is to use regression trees. While these models are inherently simpler than most of the other algorithms, their degree of interpretability is not adequate to take them as the surrogate model in economics. If they were, we would not have dedicated a chapter to understanding them.

Figure 8.8: Surrogate Models



for comprehensively understanding the relationships within the data. Without a robust theoretical framework to anchor the models, the capacity to utilize machine learning techniques to elucidate these relationships is somewhat limited. This contrasts with traditional econometric models, which are typically built on pre-existing theories regarding their functionality.

While the interpretability methods may accurately reflect how inputs influence model predictions, the emphasis in economics tends to lean toward the broader systems being represented rather than the mechanics of the models themselves. Therefore, if different models significantly distort these interpretations, and there is no theoretical underpinning to evaluate them, their utility becomes less appealing.

Interestingly, when the various methods converge on specific observations—such as consistently identifying the *cement* variable as the most important or establishing a positive relationship with sectoral indicators—this convergence may indicate that the models are effectively capturing the true dynamics of the system. Nonetheless, this conclusion hinges on the adequacy of the data used. In this context, the presence of correlated inputs poses significant challenges for interpretability, both in measuring importance and assessing the true relationship between an individual variable and the target. Although these challenges can be partially addressed through dimensionality reduction techniques like PCA to mitigate feature correlation, such solutions often introduce an additional layer of complexity to interpretation, complicating the connections back to the original variables.

However, the primary focus in nowcasting is not necessarily on understanding the intricate relationships between variables, but rather on enhancing predictive accuracy and providing explanations for those predictions. Then, despite these limitations in understanding relationships, it can be concluded that the models and interpretability techniques explored in this chapter may be

well-suited for the primary goal of nowcasting: generating more accurate predictions and providing meaningful explanations for those predictions. This suggests that machine learning approaches can be beneficial for the nowcasting framework, enhancing the ability of researchers and practitioners to make informed decisions based on timely and accurate economic forecasts.

Chapter 9

Dealing with Revisions

Revisions in economic data are an integral aspect of the nowcasting process, as they directly influence the accuracy and reliability the predictions. The importance of these revisions lies in the continuous availability of new data and the periodic updates to previously published estimates: as new information becomes accessible, the predictions made using earlier data may be rendered less relevant or entirely inaccurate, necessitating a reassessment of the nowcasting outputs.

Previous studies, such as Banbura *et al.* (2013) and Hayashi & Tachi (2021), have explored the implications of revisions in the context of GDP nowcasting. These investigations have predominantly focused on developing methodologies that assess the impact of new data and updates on existing forecasts. In contrast, this chapter seeks to address a different but critical question: how do revisions affect the certainty surrounding our GDP predictions?

The presence of revisions introduces an inherent uncertainty into the nowcasting framework, as updates in data lead to modifications in predictions. This chapter will propose a framework for modeling the uncertainty surrounding predictions due to data revisions. Particularly, it will build on the observed behavior of revisions to understand how the uncertainty around an explanatory variable translates into the uncertainty around the GDP prediction.

9.1 The methodological framework

Let us define an observation of a generic explanatory variable x for period t as x_t. A vintage refers to the moment in time when this observation was computed; let us denote an observation (or *estimate*) of x_t in the vintage v

as $x_{t,v}$, where v can be expressed as an ordinal number $1, 2, \ldots, V$, with V representing the final version of this estimate ($\lim_{v \to V} x_{t,v} = x_{t,V}$).

The key idea of this framework is that $x_{t,v}$ is the best estimate of $x_{t,V}$ available at time v, and that the final estimate can be characterized as a random variable whose value equals the estimate at time v plus a stochastic error term $\epsilon_{t,v}$. This can be expressed as:

$$x_{t,V} = x_{t,v} + \epsilon_{t,v}$$

where the error decreases to 0 in its latest version ($\epsilon_{t,V} = 0$). Furthermore, if we assume that there is no systematic bias in the estimates, $E[\epsilon_{t,v}] = 0$. The final value $x_{t,V}$ can then be characterized as following an unknown distribution F_x with mean $x_{t,v}$ and standard deviation $\sigma_{x,v}$, where $\sigma_{x,v}$ is assumed to be constant across different values of t, but specific to a given v. This yields:

$$x_{t,V} \sim F_x(x_{t,v}, \sigma_{x,v})$$

Since our variable of interest, y_t, is a function of the explanatory variables ($y_t = f(x_t)$), the final estimate will also be vulnerable to the revisions of x_t. It follows that this additional dimension of uncertainty will also be reflected in the outcome variable:

$$y_{t,v} = f(x_{t,v})$$

The question then is how to leverage the information from the vintages of x_t to learn about the uncertainty of our estimates of y_t. Our approach is to translate the possible realizations of $x_{t,V}$, given $x_{t,v}$, into a distribution of the outcome variable y_t through simulations.

9.2 Application on pseudo-synthetic data

Since vintage data for Argentinian time series are not available in my sources at the time of this study, identifying the empirical distributions of the data around their estimates at different points in time is not feasible. However, to illustrate the framework, a synthetic distribution can be assumed.

Let us assume that we have three vintages for some of our variables, $\mathbf{V} = 1, 2, 3$, where the last vintage represents the final estimate of the indicator ($V = 3$). Furthermore, let us assume that the final estimate of the variables x_V follows a vintage-dependent normal distribution around the value of the

estimate at vintage v x_v, as follows:

$$x_{V|v} \sim N(x_v; \sigma_v^2)$$

Note that we drop the time subscript and the feature subscript to make the equations simpler. Without loss of generality, we are referring to the uncertainty around one feature an the estimation of one data point. Lastly, let us assume σ_1, σ_2 and σ_3 are 0.3, 0.15 and 0, respectively. $\sigma_3 = 0$ reflects that it is the final estimate and there is no revision-related uncertainty anymore.

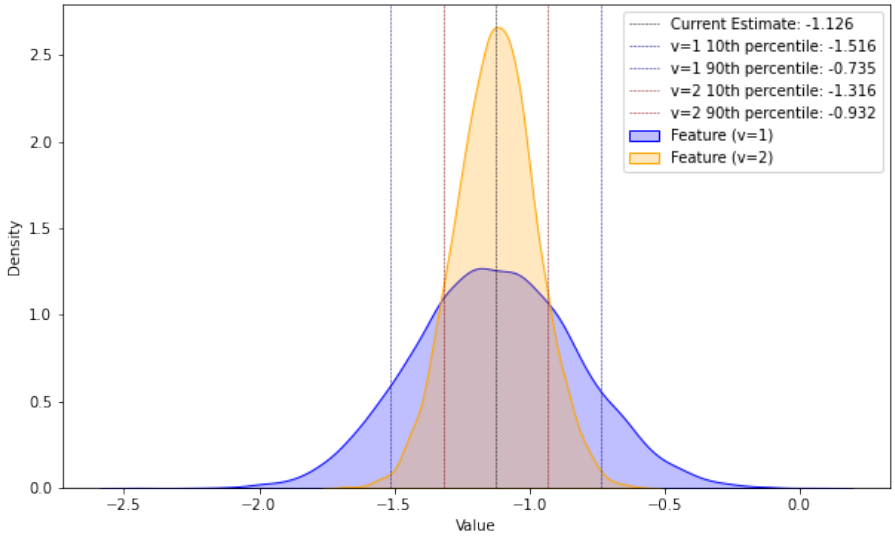
This distribution is illustrated in Figure 9.1 for the *cement* variable. In the figure, the value of the variable in the December 2023 observation is taken. By assuming that the observed value of -1.126 is the estimate of the variable at a given vintage, the possible values of the final version ($V = 3$) can be obtained by simulation. First, if we assume the observed value corresponds to vintage $v = 1$, where $\sigma_1 = 0.3$, the distribution of the final estimate is plotted in blue. The distribution is centered around the current estimate of the variable -1.126, and ranges roughly between -2.25 and 0. In a second step, let us assume the estimate at $v = 2$ is the same value. Now, the uncertainty around the final estimate is reduced, as $\sigma_2 = 0.15$. Through new simulations, the new distribution of possible values is obtained and the results are plotted in orange. The new distribution is considerably narrower, ranging roughly from -1.75 to -0.5.

Our approach involves simulating 10,000 realizations of the final values of the variables conditioned on the available information $x_{V|v}$ and computing the final outcome for each realization. This simulation yields a distribution of the output conditional on the current vintage, y_v, which captures the uncertainty stemming from the revisions of the explanatory variables. As time progresses and more accurate estimates of the explanatory variables are utilized for prediction, the uncertainty surrounding the final value of y_V is expected to decrease consistently.

It is important to note that this simulation process can be applied to both individual variables and groups of variables. This flexibility provides a comprehensive view of the uncertainty in the predictions, as well as insight into the contributions of each variable to the overall uncertainty. However, caution must be exercised in interpreting these contributions, as the uncertainty associated with each feature may not necessarily be additive.

Figure 9.2 illustrates the results of this exercise, in which we assumed that

Figure 9.1: Uncertainty Around the Estimate - Cement



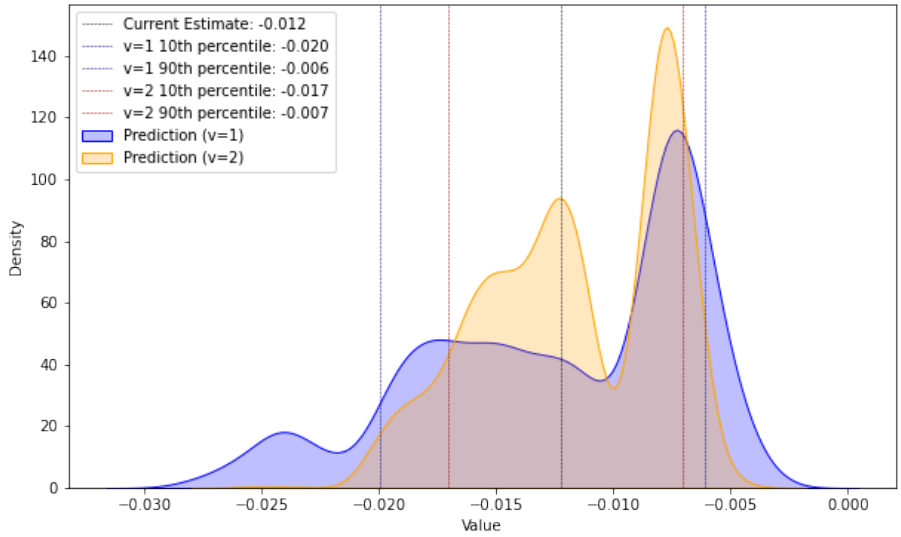
10 variables in our dataset followed the specified revision dynamics, using the Random Forest model. In this figure, 10,000 combinations of feature estimates were computed, with their corresponding predictions plotted in blue for the first vintage and in orange for the second vintage.

The resulting distributions are notably non-normal, reflecting the non-linear effects that each of the variables has on the target variable. The current prediction, which stands at -0.012, is derived from applying the currently observed input values through the model. Under the assumption of no systematic biases, this value serves as our conditional expectation for GDP in this instance. Given that the distributions for the vintages are synthetic, further analysis of these results is not warranted. However, if the true distributions were available, researchers could utilize this technique to assess the uncertainty surrounding their predictions effectively.

9.3 On decomposing the impact of revisions

While this study presents a technique for assessing the uncertainty that revisions bring to the predictions, it does not address the decomposition of the new forecast into the old forecast plus revision. The methods proposed by Banbura et al. (2013) and Hayashi and Tachi (2021) rely on the nowcasts being produced by a linear Kalman filter, allowing for an additive decomposition. This is

Figure 9.2: Uncertainty Around the GDP Prediction



not feasible, however, when the predictions are produced using more complex machine learning algorithms like the ones in this study. The answer to this question is left for further studies along with the question of how much each variable adds to the overall uncertainty. Identifying the contribution of each variable then becomes a crucial step towards reducing the uncertainty revolving the final prediction.

Chapter 10

Conclusions

In this paper, the application of various machine learning models for GDP nowcasting was explored, with a specific focus on their predictive accuracy and interpretability. The results obtained from fitting these models provide partial support for the existing literature, affirming that increased accuracy in forecasting economic indicators can be yielded by machine learning techniques. Additionally, several interpretability techniques were examined, which proved useful for addressing some questions related to the predictions but not all. In particular, it was found that these methods align well with the specific questions pertinent to nowcasting, which primarily revolve around the generation of accurate forecasts. Furthermore, a framework for understanding the impact of revisions on these predictions was proposed, adding another layer of insight to the nowcasting process.

These findings suggest that central banks would benefit from the incorporation of machine learning models into their suite of nowcasting tools. The predictive capabilities of these models can be enhanced, providing more timely and accurate assessments of economic conditions. However, it should be noted that policymakers should remain mindful of the limitations inherent in these models, particularly concerning interpretability and the complexities of underlying relationships within the data. By doing so, the strengths of machine learning can be leveraged by central banks while maintaining a critical awareness of the challenges involved in economic modeling.

Bibliography

Abd El-Aal, M., S. Abeelminaamb, & A.-E. Abd-Elatif (2023): "Nowcasting egypt gdp using machine learning algorithms." *Journal of Computing and Communication* **2**: pp. 1–8.

Babii, A., E. Ghysels, & J. Striaukas (2022): "Machine learning time series regressions with an application to nowcasting." *Journal of Business Economic Statistics* **40(3)**: pp. 1094–1106.

Banbura, M., D. Giannone, M. Modugno, & L. Reichlin (2013): "Chapter 4 - now-casting and the real-time data flow." In G. Elliott & A. Timmermann (editors), "Handbook of Economic Forecasting," volume 2 of *Handbook of Economic Forecasting*, pp. 195–237. Elsevier.

Cepni, O., I. E. Guney, & N. R. Swanson (2019): "Nowcasting and forecasting gdp in emerging markets using global financial and macroeconomic diffusion indexes." *International Journal of Forecasting* **35(2)**: pp. 555–572.

Dauphin, J.-F., K. Dybczak, M. Maneely, M. T. Sanjani, N. Suphaphiphat, Y. Wang, & H. Zhang (2022): "Nowcasting gdp - a scalable approach using dfm, machine learning and novel data, applied to european economies." *IMF WORKING PAPERS* **(2022/052)**.

Fan, J. (2019): *Real-time GDP nowcasting in New Zealand : an ensemble machine learning approach*. Master's thesis, Massey University.

Fornaro, P. & H. Luomaranta (2020): "Nowcasting finnish real economic activity: a machine learning approach." *Empirical Economics* **58(1)**: pp. 55–71.

Hayashi, F. & Y. Tachi (2021): "The nowcast revision analysis extended." *Economics Letters* **209**: p. 110112.

Hopp, D. (2024): "Benchmarking econometric and machine learning methodologies in nowcasting gdp." *Empirical Economics* **66(5)**: pp. 2191–2247.

Jonsson, K. (2020): "Machine learning and nowcasts of swedish gdp." *Journal of Business Cycle Research* **16(2)**: pp. 123–134.

Kohlscheen, E. (2021): "What does machine learning say about the drivers of inflation?" *BIS Working Papers* **(980)**.

Loermann, J. & B. Maas (2019): "Nowcasting us gdp with artificial neural networks." *MPRA Paper* **(95459, University Library of Munich, Germany)**.

Malik, N. & B. Agarwal (2022): "Time series nowcasting of india's gdp with machine learning." In "2022 International Conference on Artificial Intelligence of Things (ICAIoT)," pp. 1–6.

Marcellino, M. & V. Sivec (2021): "Nowcasting growth in a small open economy." *National Institute Economic Review* **256**: pp. 127–161.

Molnar, C. (2024): *Interpretable Machine Learning*.

Park, S. & J.-S. Yang (2022): "Interpretable deep learning lstm model for intelligent economic decision-making." *Knowledge-Based Systems* **248**: p. 108907.

Ranjan, A. & S. Ghosh (2021): "A machine learning (ml) approach to gdp nowcasting: An emerging market experience." - .

Richardson, A., T. Mulder, & T. l Vehbi (2018): "Nowcasting new zealand gdp using machine learning algorithms." *International Journal of Forecasting* .

Soybilgen, B. & E. Yazgan (2021): "Nowcasting us gdp using tree-based ensemble models and dynamic factors." *Computational Economics* **57(1)**: pp. 387–417.

Tiffin, A. J. (2016): "Seeing in the dark: A machine-learning approach to nowcasting in lebanon." *IMF WORKING PAPERS* **(2016/056)**.

Zhang, Q., H. Ni, & H. Xu (2023): "Nowcasting chinese gdp in a data-rich environment: Lessons from machine learning algorithms." *Economic Modelling* **122**: p. 106204.