MASTER'S DEGREE

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

DISSERTATION

CREATING A LARGE DATA SET FOR FORECASTING INFLATION IN THE EUROPEAN CONTEXT USING MACHINE LEARNING TECHNIQUES

VASCO COSTA LEAL

OCTOBER 2024

Lisbon School of Economics & Management
Universidade de Lisboa

MASTER'S DEGREE

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

DISSERTATION

CREATING A LARGE DATA SET FOR FORECASTING INFLATION IN THE EUROPEAN CONTEXT USING MACHINE LEARNING TECHNIQUES

VASCO COSTA LEAL

SUPERVISION:

PROF. ADRIANA CORNEA-MADEIRA

OCTOBER 2024

Abstract

Inflation forecasting plays a critical role in macroeconomic analysis due to its impact on economic policies and decision-making, which in turn could lead to economic growth. This dissertation contributes to the field by creating and introducing the European Union Economic Data (EUED) dataset, a macroeconomic dataset built upon aggregating multiple Eurostat sources, tailored for the European context, specifically for the regions European Union of 27, Euro Area of 20 and Germany. This dissertation uses the newly built EUED dataset to explore the application of machine learning techniques to forecast inflation across the regions through the macroeconomic indicator Harmonized Index of Consumer Prices (HICP) for the component All-items.

The findings demonstrate that machine learning models, in particular LASSO, consistently outperform the traditional benchmark Random Walk and Autoregressive model in terms of predictive accuracy, as it delivers the smallest predictions errors across the three metrics considered – root mean squared error, mean absolute error and median absolute deviation. Additionally, the results highlight that forecasting inflation measured in monthly rate of change yields the most reliable results across models when compared to when it is measured as an index or an annual rate of change.

Keywords:

Inflation, Forecasting, HICP, EUED, Machine Learning, LASSO, Medallion Architecture, Recursive forecast.

Acknowledgements

The completion of this dissertation would not have been possible without the support and contribution of several people to whom I owe my deepest gratitude.

First and foremost, I extend my sincere appreciation to Professor Adriana Cornea-Madeira for her invaluable guidance and thoughtful insights throughout this work.

To my girlfriend, Ilona, for her patience, love, and care. Thank you for being my right hand and constantly reminding me of the importance of pursuing and fighting for my goals. Your encouragement and unwavering support made this journey possible.

To my mother, Luisa, my father, Vasco, and my grandfather, Carlos, for being a constant source of inspiration and strength and for always standing by my side. Your belief in my potential and your unconditional support have been a cornerstone of my success.

Finally, to my friends, for the engaging conversations, shared knowledge, and companionship that made this experience even more meaningful.

To all of you, thank you for making this milestone in my life a reality.

Table of Contents

List of Tables

1. **Introduction**

Inflation can be defined as “the gradual loss of purchasing power, reflected in a broad rise in prices for goods and services” (McKinsey & Company, 2024). This means that with the same amount of money, it is not possible to buy the same amount of goods and services as before (European Central Bank, n.d.).

The motivation for this study lies in the importance of forecasting inflation due to its uncertainty. Forecasting inflation is a critical task in macroeconomic analysis with far-reaching implications for policy decisions, business strategies and consumers behavior (Gafurdjan, 2024).

Accurate and timely inflation predictions are recommended to support central banks and governments adjust their policies in order to help them pursue price stability, which in turn creates an environment that is favorable for short and long term economic growth (Mandeya & Ho, 2021).

A key inspiration for this work comes from the article *"Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods"* from Medeiros *et al.* (2019). These authors utilize the Federal Reserve Economic Data (FRED) monthly datasets of United States (US) macroeconomic indicators to forecast the Consumer Price Index (CPI). They demonstrate that machine learning methods can outperform traditional econometric models in predicting inflation, highlighting the value of leveraging data-rich environments for more accurate forecasts. Therefore, this dissertation will not only continue but will also be grounded on their work. Thus, this study contributes to literature in three ways.

First, while the previously mentioned paper focuses on forecasting US inflation through the Consumer Price Index (CPI), our analysis shifts the focus to the European context by forecasting the Harmonized Index of Consumer Price (HICP). The HICP is a more relevant measure for Europe because it gives a “comparable measure of inflation as they (prices) are calculated according to harmonized definitions” (European Union/Eurostat, 2018).

Second, our approach involves the construction of a comprehensive and unified macroeconomic dataset for Europe by aggregating multiple data sources from Eurostat

(n.d.). This dataset serves as a foundation not only for this study, but also as a valuable resource for future research on European inflation and broader macroeconomic trends.

Third, this study conducts a comparison of traditional econometric models and a variety of machine learning techniques in forecasting European inflation. By evaluating the predictive accuracy of these models, we aim to identify the most effective approach for predicting the HICP inflation indicator in a European context. We also want to assess the relative performance of traditional and machine learning methods.

Building on these contributions, the results of this study provided insights into the predictive performance of various forecasting models across the three distinct regions – European Union of 27 (EU27), Euro Area of 20 (EA20) and Germany. Model performance was assessed using three evaluation metrics – root mean squared error (RMSE), mean absolute error (MAE) and median absolute deviation (MAD) – with lower metric values indicating that a given model has a higher ability to predict. Among the models considered, the LASSO model stood out as it consistently achieved lower prediction errors across regions and metrics considered, frequently ranking among the best three performing models. While Random Forest excelled at forecasting HICP all-items component measure in monthly rate of change for the region Germany, its performance varied considerably across other regions and inflation measures, reducing its overall reliability. Notably, these findings differ from the results of Medeiros *et al.* (2019), who identified Random Forest as the best model for inflation forecasting in the US context.

Additionally, this dissertation compared the average performance of models based on different measurement methods – index, monthly rate of change and annual rate of change – for the component all-item of the HICP inflation indicator. Results showed that using the monthly rate of change led to significantly better predictive accuracy, as it produced lower errors across all metrics, compared to the other measurements. The combination of LASSO and this measurement for the indicator component provided a robust and balanced approach to forecasting inflation in the European context.

The dissertation is organized as follows. Chapter 2 presents the second contribution, the dataset created, as well as the process to make it ready for our analysis. Chapter 3 exhibits the methodology used for the third contribution, the workflow in Python. Chapter

4 reviews the models used to predict inflation. Chapter 5 discusses the results. Chapter 6 concludes the dissertation.

2. Data

The dataset used in this study, which we named European Union Economic Data (can be access through 'EUED'), is a contribution to the literature on inflation forecasting in the European context. The EUED is a monthly macroeconomic dataset designed for empirical analysis in data-rich environments. We built this dataset by aggregating multiple datasets retrieved from Eurostat's "European and National Indicators for Short-Term Analysis" database (can be access through 'Eurostat'), which provides a wide range of economic indicators at the European level.

Unlike the Federal Reserve Economic Data (can be access through 'FRED') dataset used by Medeiros *et al.* (2019), which is updated in real-time, the EUED dataset was manually compiled. While this introduces limitations in terms of real-time updates, it also provides a highly customized dataset.

The EUED dataset covers the period from January 1980 to December 2023, providing 528 observations and 744 variables for each of four regions: European Union of 27 (EU27), Euro Area of 20 (EA20), Germany, and United Kingdom (UK). Each of the 744 variables represents a unique combination of an indicator component, a specific adjustment level, and a measurement unit. In other words, each indicator component can appear across several forms of adjustment and units of measurements. These adjustments can be categorized and ordered by their degree of processing, with the highest degree being seasonally and calendar adjusted (SCA), followed by seasonally adjusted (SA), calendar adjusted (CA), not seasonally adjusted (NSA), and the lowest degree being not adjusted (NA), which can be represented as SCA > SA > CA > NSA > NA. Measurements, on the other hand, might be in a variety of units, including millions of euros, index values (year *2015=100*), monthly change rates ($month_t/month_{t-1}$), annual change rates ($month_t/month_{t-12}$), balances (*e.g.*, differences between positive and negative answers), thousands of persons, and rates (*e.g.*, a population percentage). For example, the Unemployment indicator has components like total, males, females, under 25 years total and over 25 years total, with each indicator component measured in both rate and thousands of persons, where each combination is available in two degrees of adjustment, SA and NA. Similarly, Harmonized Index of Consumer Prices (HICP) indicator – the primary focus of this study – has components such as all-items, food and non-alcoholic beverages, energy and health, with each indicator component measured in

index, monthly rate and annual rate, where each combination is available in a single degree of adjustment, NA. The table below details some of these combinations along with the variable they represent in the EUED dataset.

Table 1:Defining a variable in EUED dataset

Indicator	Component	Measurement	Adjustment	Variable
HICP	All-items	Index	Not Adjusted	HICP_All_idx_NA
		Monthly rate		HICP_All_mor_NA
		Annual rate		HICP_All_anr_NA
	Energy	Index		HICP_Energy_idx_NA
		Monthly rate		HICP_Energy_mor_NA
		Annual rate		HICP_Energy_anr_NA
Unemployment	Total	Thousands of people	Not Adjusted	L_UT_1000_NA
			Seasonally Adjusted	L_UT_1000_SA
		Rate	Not Adjusted	L_UT_rate_NA
			Seasonally Adjusted	L_UT_ rate _SA
	Total - Over 25 years	Thousands of people	Not Adjusted	L_UT+25_1000_NA
			Seasonally Adjusted	L_UT+25_1000_SA
		Rate	Not Adjusted	L_UT+25_rate_NA
			Seasonally Adjusted	L_UT+25_ rate _SA

To manage and process the data efficiently, we implemented a Medallion Architecture approach (Data Bricks, n.d.), which consists of three distinct layers: bronze, silver, and gold. This layered approach allows for data processing and storage at each step, where transformations are well-organized, easily accessible, and replicable, ensuring traceability and flexibility. It also ensures that at any point in the analysis, data from a specific processing step can be revisited, adjusted, or reanalyzed. This architecture was applied individually to each region to account for differences in data availability.

- Bronze Layer: Contains the original version of the data extracted from Eurostat, without any transformation. Holds the full dataset of 528 observations and 744 variables across each of the four regions. This layer forms the foundation of the dataset in which all further transformations are applied, ensuring that the data can be revisited and analyzed in its most basic form.

- Silver Layer: The data in this layer has undergone initial filtering and cleaning steps based on the foundation established in the bronze layer, with the aim of narrowing the study's focus. These steps include limiting the timeframe, ensuring sufficient historical coverage while balancing data availability, to be from January 2000 until December 2023, reducing the number of observations from 528 to 288 for each region, and removing variables with missing values across all observations in that period. As a result of the cleaning, due to differences in data availability, the number of variables retained were different in each region. This refined silver layer dataset is more manageable and can be used as a starting point for future studies, allowing for flexibility to explore various strategies to address the remaining missing values.

- Gold Layer: Holds the final dataset used in the forecasting models, emphasizing on the quality and completeness of the data, particularly for the key indicator, the HICP for the All-items component. Additional filtering and cleaning steps were applied on top of the silver layer dataset to achieve this. First, out of the original 75 variables related to HICP (each defined as shown in Table 1.), we retained up to three variables representing the HICP All-items component in its three measurement units: index with the variable HICP_All_idx_NA, monthly rate with HICP_All_mor_NA, and annual rate with HICP_All_anr_NA. Second, to prioritize data completeness, variables with any missing values were removed. Third, when an indicator component was available in multiple adjustments forms, we retained the one with the highest degree of adjustment according to the hierarchy previously mentioned – SCA > SA > CA > NSA > NA. For example, the indicator component Unemployment Total, measured in thousands of people, is available in two adjustment forms – SA and NA – resulting in two variables – L_UT_1000_SA and L_UT_1000_NA, respectively – as shown in Table 1. In this case, we retain the SA variable, as it includes seasonal adjustments performed by

Eurostat. While it is possible to compute SA manually using NA, this process introduces additional steps and complexity, that could lead to potential errors from manual processing. Therefore, we decided to rely on Eurostat's pre-computed adjustments. These steps were carefully sequenced to prevent unintended data loss. If a higher adjustment version contained any missing value, we retained the next best adjusted version with complete data, prioritizing data availability for the analysis. Reversing this order – selecting the highest adjustment first and then removing variables with missing values – could have led to the unintended exclusion of valuable data. This gold layer dataset served as the foundation for the forecasting analysis, balancing data quality, completeness and preprocessing efficiency.

To illustrate the impact of these transformations, Table 2. provides an overview of the number of observations (obs) and variables (vars) for each region across the three layers, as well as the HICP variables available for analysis in the final dataset.

Table 2. Medallion Architecture

Layer / Dataset / Region	Bronze Original	Silver Filtered	Gold Final	HICP Variables
EU27	528 obs 744 vars	288 obs 582 vars	288 obs 164 vars	HICP_All_idx_NA HICP_All_mor_NA
EA20	528 obs 744 vars	288 obs 597 vars	288 obs 177 vars	HICP_All_idx_NA HICP_All_mor_NA
Germany	528 obs 744 vars	288 obs 727 vars	288 obs 362 vars	HICP_All_idx_NA HICP_All_mor_NA HICP_All_anr_NA
UK	528 obs 744 vars	288 obs 449 vars	288 obs 6 vars	None

As shown in the table above, the number of variables and observations varies across the bronze, silver, and gold layers for each region. Notably, the UK's final dataset contains no valid HICP variables, in other words, all UK variables related to HICP (such as HICP_All_idx_NA) were removed during the Medallion Architecture framework due to missing values. Although the original UK dataset included 744 variables, this number was reduced to 6 in the final dataset and none of which related to the HICP All-items indicator component. Therefore, the UK region will be excluded from further analysis, as its final dataset lacks the primary indicator required for this study's focus on forecasting inflation in the European context. Nevertheless, the UK data has been retained in the EUED dataset as it may provide insights for future studies.

3. Methodology

This chapter outlines the methodology applied to forecast monthly inflation within the European context, detailing the workflow that enables a comparison of traditional econometric models and machine learning techniques, as mentioned in Chapter 1. Introduction.

While the article from Medeiros *et al.* (2019) implemented their workflow in R, our workflow was developed in Python and includes steps for data splitting, data standardization, lagged variables, model selection, cross-validation for model tuning, recursive forecasting strategy, and model evaluation. It was designed to iterate across the three regions – EU27, EA20, and Germany – final's datasets obtained after applying the Medallion Architecture approach, discuss in Chapter 2. Data. This approach focuses primarily on the HICP indicator for the All-items component, which can be represented in three distinct measurement units. Each combination forms a unique target variable – HICP_All_idx_NA, HICP_All_idx_NA and HICP_All_idx_NA – as detailed in Table 2.

Considering the following model

$$\pi_{t+h} = G(\boldsymbol{x}_t) + u_{t+h}\,, \qquad t = 1, \dots, T\,, \qquad h = 1, \dots, H,$$

where, in a given region and inflation target variable, π_{t+h} is the inflation, measured in a specific unit, in month $t + h$ where h is the periods ahead (h-step-ahead forecast); $\boldsymbol{x}_t = (x_{1t}, \dots, x_{nt})'$ is a n-vector of covariates containing a large set of potential predictors (discussed in Section 3.1. Data Split), where lags of both π_t and $\boldsymbol{x}_t$ were added as later discussed in Section 3.3. Lagged Variables; the target function $G(\boldsymbol{x}_t)$ can be a single model or an ensemble of different specifications and it is later referred to in Section 3.4. Model Selection; with $G(\cdot)$ being the mapping between covariates and future inflation using a recursive forecast strategy (discussed in Section 3.6. Recursive Forecast strategy); and u_t is a zero-mean random error. Where the one-step-ahead forecast is given by

$$\hat{\pi}_{t+1} = G(\boldsymbol{x}_{t+1})$$

3.1. Data Splitting

For a specific target variable, the set of potential predictors are all the available variables in the final dataset, excluding the set of available target variables. For example, given the EU27 region, there are two available target variables (as described in Table 2.), HICP_All_idx_NA and HICP_All_mor_NA, so to forecast the inflation measured in index, the target variable is assigned to be HICP_All_idx_NA and the predictors are set to be the remaining 162 variables (164 all available variables – 2 available target variables) such as the Rate of Total Unemployment (with L_UT_rate_SA), the Index of the Total Turnover of Retail Sale of food, beverages and tobacco (with R_RSfbt_TT_idx_SCA) and the Index of Production in Construction (with C_C_Prod_idx_SCA).

After defining the target variable π and the predictors variables $\boldsymbol{x}$, the data was split into four distinct datasets: training and testing sets for both the π and $\boldsymbol{x}$. The objective of these splits is to ensure that the models are trained on historical, in-sample, data and tested on the out-of-sample future values. With this in mind, we defined the testing dataset to be 12 months, from January 2023 to December 2023 and the training dataset to range from January 2000 to December 2022. Thus, using again the region EU27 and target variable HICP_All_idx_NA as an example, the data was split according to Table 3., for each combination of region and target variable.

Table 3: Data Split for EU27 region and HICP_All_idx_NA variable

Dataset	π training	π testing	$\boldsymbol{x}$ training	$\boldsymbol{x}$ testing
Variable	Target variable HICP_All_idx_NA		162 predictor variables	
Start date	January 2000	January 2023	January 2000	January 2023
End date	December 2022	December 2023	December 2022	December 2023

3.2. Data Standardization

Since the variables in the dataset are measured in a variety of units, as detailed in Chapter 2. Data – for instance some are in millions of euros while others are in rates of change or indexes – their scales differ, potentially biasing models towards variable with

larger magnitudes. To mitigate this issue, standardization was applied, ensuring that machine learning models could accurately compare variables (Brownlee, 2020). Standardization was performed by scaling each variable independently, subtracting its mean and dividing it by its standard deviation. This transformation shifts each variable's distribution to have a mean of zero and a standard deviation of one.

To avoid data leakage – when information from the testing set is shared with the training set, which can compromise the reliability of predictive models (Rosenblatt et al., 2024) – the scaler was fitted only on the training data. The same mean and standard deviation obtained from the fitted scaler were then applied to standardize both the training and testing datasets.

The scaling transformation applied is represented by

$$X_{standardized} = \frac{(X - \mu)}{\sigma} ,$$

where X is the original value; μ and σ are, respectively, the mean and standard deviation of the training data points for a given variable.

3.3. Lagged Variables

With all variables standardized, we created lagged versions of each variable to capture temporal dynamics and improve the forecast's predictive power, following the approach used by Medeiros *et al.* (2019). While they implemented four lagged versions, we extended this to twelve lagged versions for each variable.

First, as mentioned before, each predictor x_n represents a measurement from a component of a macroeconomic indicator that may influence inflation, but its past values may also be relevant for predicting current inflation. Hence, we included lagged versions of each predictor as additional variables in the set of potential predictors $\boldsymbol{x}$. For example, if a variable x_n represents the seasonally adjusted Unemployment Total measured in thousands of people for a given month t, its first lag $x_{n,t-1}$ corresponds to the value of this variable in the previous month $t - 1$. Similarly, the twelfth lag $x_{n,t-12}$ corresponds to the value from twelve months earlier $t - 12$. This ensured that both recent and long-term past information was available as potential predictors.

Second, like predictors, past values of inflation π, specifically $\pi_{t-1}, \ldots, \pi_{t-12}$, may also be useful to predict future inflation π_{t+1}. However, to prevent data leakage from future inflation values, these lagged versions were carefully incorporated into the $\boldsymbol{x}$ training dataset (described in Table 3.). In other words, at this stage, these lagged versions of inflation were derived exclusively from the π training dataset (also described in Table 3.) and added as new variables exclusively to the $\boldsymbol{x}$ training dataset. For the $\boldsymbol{x}$ testing dataset, the lagged inflation values were computed later in the workflow using the predicted inflation values rather than actual future observations, thus avoiding data leakage. Further details on this are provided in section 3.6. Recursive Forecasting strategy.

Creating lagged variables introduced missing values at the beginning of the dataset, depending on the lag length. For instance, a lag of one month resulted in the first row having a missing value, while a lag of twelve months led to missing values in the first twelve rows. To address this, we removed rows with missing values from both π and $\boldsymbol{x}$ training datasets, shifting the training period to start 12 months later, equal to the number of lagged versions considered. Consequently, instead of starting in January 2000 as described in Table 3., the training period was updated to begin in January 2001 and end in December 2022, ensuring no missing values remain in the final dataset.

3.4. Model Selection

As mentioned, for a given region and inflation target variable, the target function is defined as $G(\boldsymbol{x}_t)$. For this function, we considered a subset of the models used by Medeiros *et al.* (2019) in their work. Two benchmark models – Random Walk (RW) and Autoregressive (AR) – were selected as traditional econometric models for comparison against the machine learning models. In the work from Medeiros *et al.* (2019), AR outperformed RW.

In addition, we included three shrinkage methods – Least Absolute Shrinkage and Selection Operator (LASSO), Ridge Regression (RR) and Elastic Net (ElNet). Medeiros *et al.* (2019) found that these shrinkage models generally provided higher accuracy compared to most other models in their study. So, we decided to include these three due to their distinct penalization approach.

Lastly, we incorporated two ensemble methods – Bootstrap Aggregating (Bagging) and Random Forest (RF). The RF model demonstrated superior performance in the Medeiros *et al.* (2019) study, by delivering the smallest errors across all their models, particularly for long horizons, though it showed some exceptions for a few short horizons. The Bagging model received less focus from Medeiros *et al.* (2019) because its results did not stand out, however, we included it to provide a linear ensemble method alongside the nonlinear method RF.

The specifications and configurations of each model are discussed in Chapter 4. Models.

3.5. Cross-Validation for Model Tuning

To parametrize the models before making out-of-sample predictions for inflation π_t, we applied a time series cross-validation method, known as walk-forward validation, on the in-sample (training) data. Our approach was motivated by Varoquaux *et al.* (2017), who highlighted that cross-validation is commonly used to select regularization parameters that maximize predictive power. They also noted that since this process can lead to overfitting – by overly tailored the model to the data – it is an essential procedure to refit the model on the available data and evaluate its accuracy on new, unseen data.

Unlike cross-sectional data, time series data has a temporal dependency, requiring cross-validation methods that preserve the sequence of observations to avoid data leakage. This ensures that each training set contains only past information relative to each prediction point.

In walk-forward validation, the training data is split into sequential time windows that expand iteratively, with each window adding a new observation to the training set and predicting the next time step. For this study, we used a two-fold walk-forward validation approach to simulate one-step-ahead predictions, represented as follows:

- Fold 1: Train on data from January 2001 to October 2022, then predict inflation for November 2022.
- Fold 2: Train on data from January 2001 to November 2022, then predict inflation for December 2022.

Given the scope of this study, this cross-validation approach was applied to tune the parameter of each model for all combinations of regions and target variables. This ensured that each model was optimized for its specific dataset characteristics. Specifically, we tuned five models – shrinkage and ensemble models – for three regions each with varying numbers of target variables (two regions include two target variables, and one region include three target variables). For each model, the objective was to find the parameters that minimize the metric Root Mean Squared Error (RMSE detailed in section 3.7. Model Evaluation) across the two defined training folds. Once optimal parameters were identified, they were used to fit the models on the full training dataset, as advice by Varoquaux *et al.* (2017).

This process involved tuning a substantial number of parameters across the various models, regions, and target variables. For space considerations and to maintain clarity and readability, we did not include the specific parameter values in the dissertation. Instead, this section focuses on describing the process underlying the parameter tuning, ensuring transparency in the approach.

3.6. Recursive Forecasting strategy

Recapping the workflow, for a given region, inflation target variable π, the gold layer dataset was split into four distinct datasets: π and $\boldsymbol{x}$ (set of potential covariates) each divided into training and testing sets. Next, each variable in these four datasets was independently standardized using the mean and standard deviation calculated from the training data points of the respective variable. Additionally, twelve lagged versions of each covariate x_n were created and added to $\boldsymbol{x}$. Similarly, twelve lagged versions of π were derived from the π training dataset and added as new predictors to the $\boldsymbol{x}$ training dataset. Once these preprocessing steps were completed, the models were built and tuned using cross-validation on the in-sample training data, as discussed in sections 3.4. and 3.5.

For forecasting, we implemented a recursive forecasting strategy, which contrasts with the direct forecasting approach described by Medeiros *et al.* (2019). This decision was motivated by the trade-off highlighted in Marcellibo *et al.* (2006), where iterated forecasts, such as those produced by a recursive approach, provide more efficient parameter estimates than the direct forecasts, however, they are susceptible to bias if the

one-step-ahead model is miss specified. Recognizing that ensuring correct model specification is inherently challenging, we designed our forecasting methodology workflow with particular attention to each step.

Our approach involved fitting a one-step-ahead model and iteratively using that model to predict out-of-sample inflation values for the twelve months in the test period, covered by the two testing datasets – from January 2023 to December 2023. At each iteration, the predicted value for the current month was used as an input for predicting the subsequent month. Since the predicted values were used in place of observations, the errors made in earlier predictions could propagate through the subsequent forecasts. The recursive forecasting proceeded as follows:

1. Initialize with the first out-of-sample inflation prediction $\hat{y}_{t+1}$, January 2023:
   - The covariates values for January 2023, denoted as $\boldsymbol{x}_{t+1}$, were extracted from the $\boldsymbol{x}$ testing dataset.
   - As mentioned before, the lagged versions of π were added iteratively to the $\boldsymbol{x}$ testing dataset. So, at this stage, in the set of predictors $\boldsymbol{x}_{t+1}$ there are no defined lagged versions of π. Thus, they were generated from the π training dataset, considering the values of $y_t, \dots, y_{t-11}$.
   - These lagged versions were added to $\boldsymbol{x}_{t+1}$.
   - The fitted model $G(\cdot)$ was applied to $\boldsymbol{x}_{t+1}$, yielding the forecasted inflation value for January 2023, $\hat{y}_{t+1} = G(\boldsymbol{x}_{t+1})$.
   - The predicted value $\hat{y}_{t+1}$ was appended to the π training dataset to be used as an input to predict inflation for the next month, February 2023.

2. Iterative Forecasting for February 2023, $\hat{y}_{t+2}$:
   - The covariates values for February 2023, $\boldsymbol{x}_{t+2}$, were extracted.
   - Lagged variables were generated using the values from $\hat{y}_{t+1}, \dots, y_{t-10}$, retrieved from the y training dataset.
   - These lagged versions were added to $\boldsymbol{x}_{t+2}$.
   - The same fitted model $G(\cdot)$ was applied to $\boldsymbol{x}_{t+2}$, yielding $\hat{y}_{t+2} = G(\boldsymbol{x}_{t+2})$.
   - The forecast $\hat{y}_{t+2}$ was appended to the y training dataset for the next iteration.

3. Repetition for the remaining months:

- The above steps were repeated iteratively for each subsequent month, up to December 2023 – $\hat{y}_{t+12}$.

This iterative process and approach to lagging ensured that future inflation values from the π testing dataset were not used during forecasting. In other words, generating iteratively the lagged versions of the inflation target variable, data leakage from future inflation values was avoided.

3.7. Model Evaluation

The workflow concludes by assessing the performance of the models for out-of-sample predictions on the testing dataset. Three metrics were used for this evaluation: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Median Absolute Deviation (MAD). Following the approach of Medeiros *et al.* (2019), these metrics are defined as follows:

$$RMSE_{m,h} = \sqrt{\frac{1}{T - T_0 + 1} \sum_{t=T_0}^{T} \hat{e}_{t,m,h}^2}$$

$$MAE_{m,h} = \frac{1}{T - T_0 + 1} \sum_{t=T_0}^{T} |\hat{e}_{t,m,h}|$$

$$MAD_{m,h} = median\left[\left|\hat{e}_{t,m,h} - median\left(\hat{e}_{t,m,h}\right)\right|\right]$$

Where $\hat{e}_{t,m,h} = \pi_t - \hat{\pi}_{t,m,h}$ and $\hat{\pi}_{t,m,h}$ is the inflation forecast for month t made by model m. Since we are doing one-step-ahead forecasts, $h = 1$.

By adopting this approach, we aimed to provide a comprehensive evaluation of model performance. RMSE and MAE are standard metrics in the forecasting literature. MAD, as mentioned by Medeiros *et al.* (2019), is valuable for its robustness to outliers and asymmetries. Reporting all three measures ensures that the results are not skewed by a few large forecasting errors.

4. Models

4.1. Benchmark

4.1.1. Random Walk (RW)

The underlying concept of a RW model is that, given the random and unpredictable nature of future movements, the current value provides the best estimate for the next time step. Therefore, the naïve forecasting approach, in which the forecast for every future period is set to the last observed value in the series, is based on the RW model (Hyndman & Athanasopoulos, 2021).

The iterative forecasting process is an one-step-ahead forecast for the twelve months in the test period – from January 2023 to December 2023 – where π_t is the last observed inflation value in the training data – for December 2022. So, the forecasts can be written as:

$$\hat{\pi}_{t+1} = \pi_t,$$

$$\hat{\pi}_{t+2} = \hat{\pi}_{t+1},$$

...

$$\hat{\pi}_{t+12} = \hat{\pi}_{t+11}$$

4.1.2. Autoregressive (AR)

In an AR model of order p, the target variable is forecasted by using its lagged values as predictors. This essentially turns the forecasting task into a multiple regression problem, where the number of predictors for the model corresponds to the order of p (Hyndman & Athanasopoulos, 2021). According to Marcellibo *et al.* (2006), the one-step-ahead AR(p) model for π_t is

$$\pi_{t+1} = \alpha + \sum_{i=1}^{p} \emptyset_i \, \pi_{t+1-i} + \varepsilon_t$$

where the parameters $\alpha, \emptyset_1, \ldots, \emptyset_p$ for iterated AR forecast are estimated recursively by Ordinary Least Squares (OLS). Thus, by setting $p = 12$, the forecasts can be constructed recursively as:

$$\hat{\pi}_{t+1} = \alpha + \phi_1 \pi_t + \phi_2 \pi_{t-1} + \cdots + \phi_{12} \pi_{t-11}$$

$$\hat{\pi}_{t+2} = \alpha + \phi_1 \hat{\pi}_{t+1} + \phi_2 \pi_t + \cdots + \phi_{12} \pi_{t-10}$$

...

$$\hat{\pi}_{t+12} = \alpha + \phi_1 \hat{\pi}_{t+11} + \phi_2 \hat{\pi}_{t+10} + \cdots + \phi_{12} \pi_t$$

4.2. Shrinkage Methods

Shrinkage models are a type of linear models designed to mitigate overfitting by applying penalties to variables with lower predictive power by shrinking their coefficients. In other words, these models introduce additional constraints to reduce the influence of less relevant variables. To implement these methods, in addition to use the work from Medeiros *et al.* (2019) and the book from Hastie *et al.* (2009) as guidance, we took inspiration from the article by García-Nieto *et al.* (2021).

Let π_t represent the value of the inflation target variable at time t, $\boldsymbol{x}_t = (x_{1t}, x_{2t}, \ldots, x_{nt})'$ denote the n-vector of covariates, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_n)'$ represent the vector of coefficients. A popular method to estimate $\boldsymbol{\beta}$ is the Ordinary Least Squares (OLS), which minimizes the residual sum of squares (RSS), that can be written as:

$$\widehat{\boldsymbol{\beta}}^{RSS} = \underset{\boldsymbol{\beta}}{argmin} \sum_{t=1}^{T} \left(\pi_t - \beta_0 - \sum_{i=1}^{n} \beta_i x_{it} \right)^2$$

In shrinkage methods, the coefficients are instead estimated by minimizing a penalized residuals sum of squared (PRSS),

$$\widehat{\boldsymbol{\beta}}^{PRSS} = \underset{\boldsymbol{\beta}}{argmin} \left\{ \sum_{t=1}^{T} \left(\pi_t - \beta_0 - \sum_{i=1}^{n} \beta_i x_{it} \right)^2 + \text{shrinkage penalty} \right\}$$

The shrinkage penalty is calculated based on the regularization (or penalty) parameter λ and on the specific type of penalty term applied. The parameter λ controls the strength of regularization: larger values increase the amount of shrinkage, while $\lambda = 0$ means that the shrinkage penalty has no effect so the estimations for the coefficients would be produced by OLS. The type of penalty term depends on the specific shrinkage model applied. In this study, we focused on the methods mentioned on the section 3.4. Model Selection: Ridge Regression (RR), Least Absolute Shrinkage and Selection Operator (LASSO) and Elastic Net (ElNet).

4.2.1. Ridge regression (RR)

Hoerl and Kennard (1970) introduced the Ridge model, which approaches the task of estimating the coefficients by minimizing PRSS:

$$L^{ridge}(\beta) = \sum_{t=1}^{T}\left(\pi_t - \beta_0 - \sum_{i=1}^{n}\beta_i x_{it}\right)^2 + \lambda\sum_{i=1}^{n}{\beta_i}^2$$

As λ increases, RR shrinks the coefficients estimates, which significantly lowers the variance of predictions, at the expense of a slight increase in bias. This shrinkage reduces overfitting and aims to improve prediction accuracy by penalizing large regression coefficients.

Even though RR shrinks all the coefficients toward zero, it does not perform variable selection because it will not set them to exactly zero for any size of $\lambda \neq \infty$ and therefore it does not enhance the interpretability of the model.

4.2.2. Least Absolute Shrinkage and Selection Operator (LASSO)

Tibshirani (1996) introduced the LASSO (short for Least Absolute Shrinkage and Selection Order) model, which estimates the coefficients by minimizing:

$$L^{lasso}(\beta) = \sum_{t=1}^{T}\left(\pi_t - \beta_0 - \sum_{i=1}^{n}\beta_i x_{it}\right)^2 + \lambda\sum_{i=1}^{n}|\beta_i|$$

Similarly to RR, LASSO also shrinks the coefficients estimates towards zero. However, unlike Ridge, LASSO's shrinkage penalty can force some coefficient estimates to become exactly zero when λ is set to be sufficiently large. This leads to variable selection, making the resulting models generally easier to interpret compared to those produced by RR.

4.2.3. Elastic Net (ElNet)

Zou and Hastie (2005) introduced the Elastic Net model, which aims to minimize:

$$L^{ENR}(\beta) = \frac{1}{2T}\sum_{t=1}^{T}\left(\pi_t - \beta_0 - \sum_{i=1}^{n}\beta_i x_{it}\right)^2 + \lambda\left(\frac{1-\alpha}{2}\sum_{i=1}^{n}\beta_i^2 + \alpha\sum_{i=1}^{n}|\beta_i|\right)$$

The ElNet model combines the shrinkage penalties from both RR and LASSO regressions. It is a convex combination of these two methods, controlled by a mixing parameter α, where $\alpha = 0$ corresponds to RR and $\alpha = 1$ corresponds to LASSO. This introduces two parameters to tune: λ and α. This model is particularly useful when the number of predictors (n) is much bigger than the number of observations (T).

4.3. Ensemble Methods

Ensemble methods are learning algorithms that combine the predictions of several base estimators built using a given learning algorithm to produce a stronger model, aiming to improve predictive accuracy over a single estimator. This dissertation relied on the work of Medeiros *et al.* (2019) as a foundational guide for developing the ensemble approach for implementing the models chosen and mentioned on the section 3.4. Model Selection: Bootstrap Aggregating (Bagging) and Random Forest (RF).

4.3.1. Bootstrap Aggregating (Bagging)

Breiman (1996) introduced the Bagging (short for Bootstrap Aggregating) model, which, in this study's case, involved the following steps:

1. Bootstrap Sampling:
   - Let $B = \{100, 150, 200, 250\}$ be the number of bootstrap samples and $b = 1, \dots, B$.
   - Were created B bootstrapped datasets by randomly sampling variables from the set of potential predictors $\boldsymbol{x}$.
   - Thus, each b dataset contained a subset of the variables, sampled independently for each bootstrapped dataset.
2. Training Base Models:
   - A Linear Regression model is trained on each b dataset, using the selected variables and the inflation target variable π.
3. Forecasting:
   - Each trained model is used to generate one-step-ahead predictions $\hat{\pi}_{t,b}$.
4. Aggregation:
   - The final prediction $\hat{\pi}_t$ is computed by averaging the predictions from all models across the B bootstrapped datasets:

$$\hat{\pi}_t = \frac{1}{B}\sum_{b=1}^{B} \hat{\pi}_{t,b}$$

4.3.2. Random Forest (RF)

Breiman (2001) introduced the Random Forest model as an extension of regression decision trees. Regression trees are models that sequentially partition the predictor space into smaller regions. This partitioning is achieved through a recursive binary splitting process, with each split chosen to minimize the sum of squared errors between observed and predicted values. This process creates a tree structure, where each terminal node corresponds to a region.

To illustrate this structure, consider the Figure 1 and the example retrieved from Medeiros *at el.* (2019), where the predictor space is defined by two variables, X_1 and X_2:

- Initial Split: The predictor space is partitioned into two regions based on the variable X_1. Observations satisfying the condition $X_1 \leq s_1$ are assigned to the region on the left, while those with $X_1 > s_1$, are assigned to the region on the right.
- Second Split: Within the region on the left ($X_1 \leq s_1$), another split occurs based on X_2. Observations are assigned to region R_1 if $X_2 \leq s_2$, and to region R_2 if $X_2 > s_2$.
- Third Split: On the right side ($X_1 > s_1$), observations are further split based on X_1. Observations are assigned to region R_3 if $X_1 \leq s_3$ and to another partition if $X_1 > s_3$.
- Final Split: The remaining observations where $X_1 > s_3$, are split based on X_2. Observations are assigned to region R_4 if $X_2 \leq s_4$ and to region R_5 if $X_2 > s_4$.

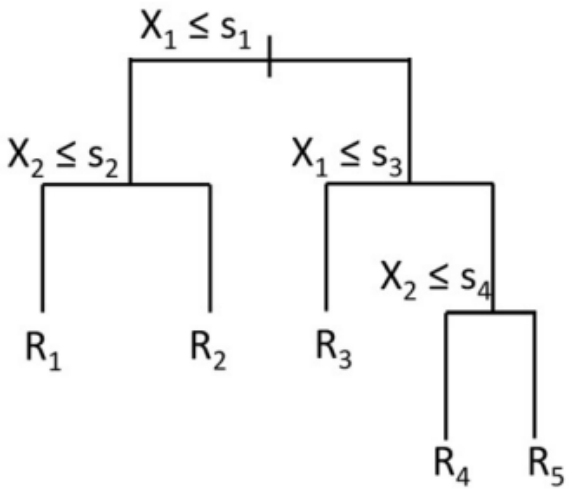

Figure 1: Example of a regression tree. Reproduction of part of Figure 1. in Medeiros et al. (2019) and part of Figure 9.2. in Hastie, Tibshirami, and Friedman (2001)

Random Forest builds upon the concept of regression decision trees and takes a step further by creating an ensemble of multiple trees like the one shown in Figure 1, where each tree is trained on a random subset of the data. This ensemble approach, constructed through the process of Bagging (as detailed in the sub-section above), aims to reduce variance and improve predictive accuracy.

To implement Random Forest, we followed steps similar to those used for Bagging, which selected a random subset of the variables from the set of potential predictors $\boldsymbol{x}$, in

addition of having the tree learning algorithm to sample from observations. The steps can be represented as:

1. Bootstrap Sampling:
   - Similarly to the model Bagging, let $B = \{100, 150, 200, 250\}$ be the number of bootstrap samples and $b = 1, \dots, B$.
   - Create B bootstrapped datasets by randomly sampling variables, from the set of potential predictors $\boldsymbol{x}$, and sampling observations.
2. Regression Decision Tree:
   - For each b dataset, train a regression tree, using the selected predictors and the inflation target variable π. Generate one-step-ahead predictions $\hat{\pi}_{t,b}$.
3. Aggregation:
   - As explained for Bagging, the final prediction $\hat{\pi}_t$ is calculated by averaging the predictions from all B trees.

Since this study involves time series data, temporal dependencies can be disrupted by standard bootstrap sampling of observations. Medeiros *et al.* (2019) addressed this by using a technique called *block bootstrapping*, which retains the temporal structure within blocks of data to preserve some of the temporal dependencies while creating the bootstrapped samples. However, in this dissertation, block bootstrapping was not implemented. Instead, the number of lagged versions of both the target variable π and covariates $\boldsymbol{x}$ we extended from 4 (as used in Medeiros *et al.* (2019)) to 12. This approach aims to capture time dependencies while still benefiting from the randomness of RF models.

5. Results and Discussion

To evaluate model's performance, three metrics were used as detailed in Section 3.7. Model Evaluation: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Median Absolute Deviation (MAD). The results are presented in the following Tables 4 to 10. Each table corresponds to a unique combination of region and target variable. Within each table, models are ranked based on RMSE, from lowest to highest. Recapping the target variables (defined as in Table 1) analyzed:

- HICP_All_idx_NA:
    - Stands for the Harmonized Index of Consumer Prices (HICP), for the component All-items (All), measured as an index that considers the base year *2015=100* (idx), and in the not adjusted degree of adjustment (NA). It reflects aggregate price level changes over time.

- HICP_All_mor_NA:
    - Denotes the monthly rate of change in HICP, capturing short-term inflation changes.

- HICP_All_anr_NA:
    - Measures the annual rate of change in HICP, offering insights into long-term inflation changes. This variable is available only for Germany in this analysis.

Table 4: EU27 results for HICP_All_idx_NA

Model	*RMSE*	*MAE*	*MAD*
LASSO	0.5648	0.4303	0.3131
Elastic Net	0.5957	0.4499	0.3077
Bagging	0.7487	0.6843	0.1803
Autoregressive	0.9775	0.7394	0.4211
Ridge	2.4092	2.111	1.0409
Random Walk	3.4399	3.1958	0.665
Random Forest	8.1125	7.707	1.9522

Table 5: EU27 results for HICP_All_mor_NA

Model	*RMSE*	*MAE*	*MAD*
Ridge	0.2797	0.268	0.1628
LASSO	0.3646	0.2845	0.25
Elastic Net	0.3646	0.2845	0.25
Random Forest	0.3888	0.3343	0.2862
Bagging	0.5278	0.4138	0.3989
Random Walk	0.5986	0.5167	0.25
Autoregressive	0.7704	0.6387	0.3766

Table 6: EA20 results for HICP_All_idx_NA

Model	*RMSE*	*MAE*	*MAD*
LASSO	0.3507	0.2536	0.1645
Elastic Net	0.3609	0.2528	0.1481
Autoregressive	0.7012	0.6202	0.4568
Bagging	1.1715	1.0319	0.9235
Random Walk	2.9113	2.6767	0.625
Ridge	3.9341	3.4008	1.8294
Random Forest	7.6716	7.2674	1.4799

Table 7: EA20 results for HICP_All_mor_NA

Model	*RMSE*	*MAE*	*MAD*
Ridge	0.3769	0.3406	0.2792
LASSO	0.4178	0.3383	0.3
Elastic Net	0.4178	0.3383	0.3
Random Forest	0.6382	0.5409	0.3903
Autoregressive	0.7315	0.6111	0.3188
Bagging	0.7558	0.6229	0.4311
Random Walk	0.7561	0.6667	0.3

Table 8: Germany results for HICP_All_idx_NA

Model	*RMSE*	*MAE*	*MAD*
Elastic Net	1.6964	1.5572	0.4529
LASSO	1.7734	1.6545	0.438
Autoregressive	2.3657	2.1925	0.2936
Random Walk	4.1069	3.8667	0.8
Bagging	5.2201	4.8527	1.5129
Ridge	5.4593	5.0542	1.5821
Random Forest	8.1768	7.6587	1.8953

Table 9: Germany results for HICP_All_mor_NA

Model	*RMSE*	*MAE*	*MAD*
Random Forest	0.4313	0.3462	0.2975
LASSO	0.5396	0.4572	0.4674
Elastic Net	0.5419	0.4578	0.494
Autoregressive	0.8552	0.6944	0.7354
Ridge	0.892	0.7525	0.3873
Ragging	1.0486	0.9278	0.3796
Random Walk	1.5927	1.5167	0.2

Table 10: Germany results for HICP_All_anr_NA

Model	*RMSE*	*MAE*	*MAD*
LASSO	1.6214	1.3837	0.6869
Bagging	1.6547	1.2619	0.4085
Random Forest	1.8216	1.1708	0.455
Ridge	1.8322	1.4529	0.4784
Autoregressive	2.4477	2.142	0.9863
Random Walk	4.1297	3.4917	1.75
Elastic Net	4.2378	3.4825	1.73

The analysis in this study focuses on comparing models, target variables, and metric, excluding comparisons between regions. Following this train of thoughts, it was possible to retrieve a few insights from Tables 4 to 10.

- HICP_All_idx_NA (index):

Since a model is better in terms of performance than another if its value for a specific metric is lower, we can say that, for this target variable, the models LASSO and Elastic Net consistently outperformed other models across the three performance metrics because they showed results with the lowest metrics values. In contrast, the Random Forest model performed significantly worse than the rest.

- HICP_All_mor_NA (monthly rate):

For the aggregate regions EU27 and EA20, the Ridge model appeared as the best performing model, followed by LASSO and Elastic Net. For Germany, the Random Forest model outperformed all the other models. The worst model candidates in terms of overall performance on the three metrics were the two benchmarks (Random Walk and Autoregressive) and the Bagging. An exception occurred for the metric MAD in Germany, where the Random Walk model delivered the best result, the lowest metric value.

- HICP_All_anr_NA (annual rate):

For his variable, available exclusively for Germany, LASSO and Bagging models achieved the best results in terms of performance. On the other hand, Random Walk and Elastic Net were the worst performing models.

Table 11 summarizes the frequency in which each model was ranked the best or worst in term of performance across metrics. While this table provided an overall view of model performance, it should not be used as the sole basis for selecting the best and worst model overall. For instance, the Ridge model was ranked the best performing model five times in total and only ranked the worst once, however when not at the top, it tended to rank closer to the bottom, as can be seen in Tables 4 to 10.

Table 11: Summary statistics for metrics

Model	*#Min. RMSE*	*#Min. MAE*	*#Min. MAD*	*Count Min*	*#Max. RMSE*	*#Max. MAE*	*#Max. MAD*	*Count Max*
RW			1	1	2	3	1	6
AR			1	1	1	1	1	3
RR	2	1	2	5			1	1
LASSO	3	1		4				0
ElNet	1	3	1	5	1			1
Bagging			2	2			2	2
RF	1	2		3	3	3	2	8

Finally, the Table 12 below highlighted that, on average, models achieved considerably better predictive accuracy (lower RMSE, MAE and MAD) when forecasting inflation measured in monthly rates of change through the variable HICP_All_mor_NA.

Table 12: Average result metrics

Region	Target variable	Avg. RMSE	Avg. MAE	Avg. MAD
EU27	HICP_All_idx_NA	2.4069	2.1882	0.6972
EU27	HICP_All_mor_NA	0.4706	0.3915	0.2821
EA20	HICP_All_idx_NA	2.4430	2.2148	0.8039
EA20	HICP_All_mor_NA	0.5849	0.4941	0.3313
Germany	HICP_All_idx_NA	4.1141	3.8337	0.9964
Germany	HICP_All_mor_NA	0.8430	0.7361	0.4230
Germany	HICP_All_anr_NA	2.5350	2.0551	0.9279

Based on the results presented in Tables 4 to 12 and by synthesized them, it was possible to draw some practical takeaways regarding the best performing model overall and target variable preference. Additionally, the findings were compared with the results obtained by Medeiros *et al.* (2019), with a focus on evaluating similarities and differences in outcomes across methodologies, regions, and periods.

- Winning Model

Across metrics, target variables and regions, the LASSO model consistently showed strong predictive accuracy. In addition, it was ranked four times as the best performing and never ranked as the worst (Table 11) and, even when the LASSO model was not the best performing, it tended to be ranked on the top three (Table 4 to 10).

Ridge and Elastic Net models were also suitable candidates. Ridge for its robust performance in forecasting inflation on the aggregate regions EU27 and EA20 using the target variable HICP_All_mor_NA. Elastic Net for being toe-to-toe with LASSO.

Although Random Forest outperformed the rest in the case for Germany with the target variable HICP_All_mor_NA, its inconsistent performance across other metrics, targets variables and regions limits its applicability to be the winning model.

Thus, LASSO could be considered as the winning model for predicting inflation across various regions and measurements.

- Wining Target Variable

As shown in Table 12, forecasting inflation through the measurement monthly rate of change resulted in considerably lower RMSE, MAE and MAD average values, suggesting that models were able to predict inflation with better accuracy. Thus, the winning target variable is HICP_All_mor_NA.

- Combination

LASSO paired with HICP_All_mor_NA can be considered the most balanced approach for forecasting inflation in the European context, considering this dissertation's specific regions, data (including time period), models and methodology.

Aside from analyzing the results in the context of this dissertation, we decided to compared them to the findings of Medeiros *et al.* (2019). While there are notable differences in scope, such as their focus on US data and different time periods, certain insights can be taken.

In both studies, the shrinkage methods were able to produced more accurate forecast than the standard benchmarks. This highlighted the benefits of machine learning methods in a data rich environment for macroeconomic forecasting, in particular for inflation.

Random Forest emerged as the winning model in Medeiros *et al.* (2019), delivering the smallest errors for most horizons. Its superiority was even more pronounced in scenarios with high inflation volatility and when considering long horizons, being slightly outperformed by shrinkage methods for short horizons. However, in this study, Random Forest showed less consistent performance, excelling only for Germany with the target variable HICP_All_mor_NA but was ranked the worst eight times in total across the three metrics. The discrepancy of these results may stem from differences in data availability on the datasets EUED and FRED, in time periods from 2000-2023 to 1990-2015 and in methodologies. For example, this study extended the number of lagged versions of all variables to twelve instead of considering block bootstrapping. In addition, Medeiros *et al.* (2019) found that shrinkage outperformed Random Forest for short horizons, but we applied a recursive one-step-ahead forecasting strategy, while they used a direct h-steps-ahead forecasting strategy.

6. Conclusion

This dissertation's main goal was to apply machine learning methods to forecast monthly inflation forecasting within the European context, focusing on EU27, EA20, and Germany. To achieve this, we built the EUED database, developed the forecasting workflow in Python, and applied well established predictive models.

The findings from this dissertation demonstrate the potential of machine learning techniques for macroeconomic forecasting. Among the evaluated models, the LASSO model showed superior consistency in predictive accuracy across metrics, target variables and regions. Elastic Net and Ridge models also delivered robust performance, while Random Forest excelled only in an isolated case. Overall, the machined learning models outperformed the benchmark models Random Walk and Autoregressive, highlighting the benefits of machine learning methods in a data rich environment for macroeconomic forecasting.

One key insight from this analysis is the importance of selecting the appropriate target variable. Forecasting inflation using the macroeconomic indicator Harmonized Index of Consumer Prices (HICP) for the All-items component produced considerably lower errors when the indicator component was measured as a monthly rate of change rather than as an index or an annual rate of change.

Apart from the methodological contributions addressed in this work, the creation of the EUED dataset stands out as one of the most noteworthy and lasting achievements of this research. This dataset represents a valuable resource for future research in the field of macroeconomics as it consists in an aggregation of multiple datasets from macroeconomic Eurostat sources. Most efforts were invested in building this database and ensuring data quality through data engineering techniques, like the approach of Medallion Architecture, to make it ready to be the backbone used for forecasting models. This structured approach allows future researchers to implement customized data transformations according to their needs.

Nevertheless, even though this study provides both the database and the Python workflow, certain limitations should be acknowledged. The database is not updated in real time, which restricts its utility for dynamic decision-making. A significant challenge involved the implementation of cross-validation due to its time complexity. While we

aimed to use a 12 time-window cross-validation setup to simulate the recursive strategy of 12 one-steps out-of-sample predictions, we discovered a coding error close to the deadline that required rerunning the entire process to get the corrected results. Given the long model training times, we had to adjust the cross-validation procedure to fit the schedule.

Future studies could address these limitations in several ways. Incorporating real-time data updates from Eurostat into the EUED dataset would significantly enhance its utility for real time applications, as the dataset will otherwise become outdated over time for analysis that required timely decisions. Different preprocessing strategies could also be explored within the gold layer, such as testing alternative methods for handling the remaining missing values from the silver layer rather than removing the entire variables. Regarding the various adjustment forms, another preprocessing approach could involve retaining variables that represent the indicator components in their NA form and, when this form is not available, manually detrending and removing seasonal components from another available adjustment series could be considered rather than retaining the highest degree of adjustment with complete data, aiming to improve data comparability by having the same degree of adjustment across variables. Furthermore, exploring direct multi-step forecasting strategies and comparing them with the recursive approach used in this work would be an interesting starting point for future studies, as it could provide valuable insights into their respective strengths and limitations. Revisiting Random Forest model with block bootstrapping could improve this model robustness. Additionally, incorporating more computationally efficient cross-validation schemes could enhance overall model robustness and performance.

This dissertation has demonstrated the efficacy of machine learning methods for inflation forecasting while providing a framework and a dataset that lay a solid foundation for future research. We hope that this work contributes to advancing the development of more accurate, scalable, and reliable macroeconomic forecasting models.

References

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*, 1937-1967. https://doi.org/10.1007/s10462-020-09896-5

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24, 123–140.

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

Brownlee, J. (2020). *How to Use StandardScaler and MinMaxScaler Transforms in Python*. Machine Learning Mastery. https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/

Dantas, T. M., & Oliveira, F. L. C. (2018). Improving time series forecasting: An approach combining bootstrap aggregation, clusters and exponential smoothing. *International Journal of Forecasting*, 34(4), 748-761. https://doi.org/10.1016/j.ijforecast.2018.05.006

Data Bricks. (n.d.). *What is a medallion architecture?*. https://www.databricks.com/glossary/medallion-architecture

European Central Bank. (n.d.). *What is inflation?*. https://www.ecb.europa.eu/ecb-and-you/explainers/tell-me-more/html/what_is_inflation.en.html

European Union/Eurostat. (2018). Harmonised Index of Consumer Prices (HICP). *Methodological Manual*, 360.

Eurostat. (n.d.). *Euro Indicators Database*. https://ec.europa.eu/eurostat/web/euro-indicators/database

Gafurdjan , Z. (2024). INFLATION AND ITS EFFECTS ON CONSUMER BEHAVIOR AND ECONOMIC POLICIES. *QO‘QON UNIVERSITETI XABARNOMASI*, *10*(10), 3–6. https://doi.org/10.54613/ku.v10i10.895

García-Nieto, P. J., Garcia-Gonzalo, E., & Paredes-Sánchez, J. P. (2021). Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-net machine learning techniques. *Neural Computing and Applications*, *33*, 17131-17145. https://doi.org/10.1007/s00521-021-06304-z

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts. Forecasting: Principles and Practice (3rd ed)

Mandeya, S. M. T., & Ho, S. Y. (2021). Inflation, inflation uncertainty and the economic growth nexus: An impact study of South Africa. *MethodsX*, *8*. https://doi.org/10.1016/j.mex.2021.101501

Marcellino, M., Stock, J. H., & Watson, M. W. (2006). A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of econometrics*, 135(1-2), 499-526. https://doi:10.1016/j.jeconom.2005.07.020

McKinsey & Company. (2024). *What is inflation?*. https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-inflation

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., & Zilberman, E. (2019). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, *39*(1), 98-119. https://doi.org/10.1080/07350015.2019.1637745

Rosenblatt, M., Tejavibulya, L., Jiang, R., Noble, S., & Scheinost, D. (2024). Data leakage inflates prediction performance in connectome-based machine learning models. *Nature Communications*, *15*(1). https://doi.org/10.1038/s41467-024-46150-w

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society*, 58, 267–288.

Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Thirion, B. (2017). Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. *NeuroImage*, 145, 166-179. https://doi.org/10.1016/j.neuroimage.2016.10.038

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society*, Series B, 67, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x