Andreas Engebretsen
Magnus Mørk

Bonding with XAI: Exploring the Potential for Sovereign Bond Spread Predictions

TIØ4900 - Financial Engineering, Master's Thesis

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Morten Risstad
June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Andreas Engebretsen
Magnus Mørk

Bonding with XAI: Exploring the Potential for Sovereign Bond Spread Predictions

TIØ4900 - Financial Engineering, Master's Thesis

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Morten Risstad
June 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

As we, Andreas Engebretsen and Magnus Mørk, conclude our Master of Science degree in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU) in the spring of 2023, we present this thesis. The study investigates the possibilities of predicting sovereign bond spreads using machine learning and explainable artificial intelligence. It represents a fully independent work conducted by us.

Our motivation for this thesis is rooted in our passion for explainable artificial intelligence and the global economic landscape. We believe that transparency in financial systems is essential to ensure safe, sustainable, and efficient markets. We recognize and appreciate the work of those preceding our own and hope to contribute insightful and productive observations to the existing body of research. We wish to express our gratitude to our academic advisors at NTNU, Professor Sjur Westgaard and Associate Professor Morten Risstad, for their invaluable guidance throughout the project.

Abstract

This study investigates the feasibility and accuracy of using Machine Learning (ML) and Explainable Artificial Intelligence (XAI) methods to predict sovereign bond spreads. The study evaluates AI models, specifically Artificial Neural Network (ANN) and Light Gradient Boosting Machine (LightGBM), compared to econometric benchmarks. The Shapley Additive Explanations (SHAP) framework is used to provide explainability for the AI models. The analysis uses a quarterly dataset from Greece, Italy, Portugal and Spain (GIPS), spanning the years 1999 to 2020. The test set covers the period from 2017 to 2020.

The objective of this research is twofold: (1) to assess the out-of-sample predictive accuracy of artificial intelligence models in forecasting sovereign bond spreads and (2) to evaluate the interpretability of these models. Additionally, the study examines the importance of the variables applied.

Our findings demonstrate that ML paired with Explainable AI (XAI) is well-suited for predicting bond spreads, delivering both accuracy and explainability. The LightGBM model performs similarly to the benchmark models, whereas the ANN model performs worse in terms of conventional regression metrics. Regarding directional classification accuracy, both AI models surpass the benchmarks. Additionally, both models offer meaningful indications concerning the importance of variables, with the LightGBM model providing the clearest interpretability. The most important exogenous variables identified across models and countries, namely GDP per capita, GDP growth, CPI, government debt, and unemployment, align with the existing literature in the field.

Reliable predictions of bond spreads can allow policymakers to take proactive steps to prevent economic challenges, which motivates this research. This research fills a gap in the existing literature by being the first, to the best of the authors' knowledge, to explore the application of ML and XAI techniques for predicting sovereign bond spreads. By conducting a comprehensive comparison of AI and benchmark models, the study provides insights into the potential of AI in this domain.

Sammendrag

Denne studien undersøker bruk av maskinlæring (ML) og forklarbar kunstig intelligens (XAI) for å predikere spredningen på statsobligasjonsrenter. Studien sammenligner AI-modeller, spesifikt ANN og LightGBM, med økonometriske referansemodeller. For å hente forklarbarhet fra AI-modellene blir SHAP-rammeverket benyttet. Det blir brukt et kvartalsvis datasett fra Hellas, Italia, Portugal og Spania (GIPS), fra årene 1999 til 2020. Testsettet dekker tidsperioden fra 2017 til 2020.

Målet med denne studien er todelt: (1) å vurdere prediksjonsnøyaktigheten av AI modeller i prediksjon av spredningen på statsobligasjoner og (2) å evaluere tolkbarheten av resultatene. Videre undersøker studien hvilke variabler som påvirker prediksjonene.

Funnene viser at maskinlæring kombinert med forklarbar AI er godt egnet for å predikere obligasjonsspredninger, og gir både presise prediksjoner og forklarbarhet. LightGBM-modellen viser en prediktiv evne som er jevn med referansemodellene, mens ANN-modellen viser en lavere nøyaktighet knyttet til størrelsesordenen på prediksjonene. Når det gjelder retningsbestemt klassifisering, overgår begge AI-modellene referansemodellene. Begge modellene gir også meningsfull forklarbarhet, der LightGBM-modellen gir de tydeligste resultatene. De viktigste variablene identifisert på tvers av modellene er BNP per innbygger, BNP-vekst, KPI, statsgjeld og arbeidsledighet. Dette stemmer overens med den eksisterende litteraturen.

Pålitelige prediksjoner av rentespredninger vil gi beslutningstakere muligheten til å unngå økonomiske utfordringer, noe som motiverer denne forskningen. Denne studien bidrar til å fylle en mangel i den eksisterende forskningen ved å være den første til å utforske bruk av XAI-teknikker for å predikere spredningen på statsobligasjonsrenter. Studien gir innsikt i AI's potensiale på dette området ved å sammenligne AI modeller med referansemodeller.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACF Autocorrelation Function.

ADF Augmented Dickey-Fuller.

ANN Artificial Neural Network.

API Application Programming Interface.

AR Autoregressive.

ARIMA Autoregressive Integrated Moving Average.

ARIMAX Autoregressive Integrated Moving Average with Exogenous Variables.

BNN Bayesian Neural Network.

CLR Cyclical Learning Rate.

CPI Consumer Price Index.

DART Dropout meet multiple Additive Regression Trees.

ELI5 Explain Like I'm 5.

ELU Exponential Linear Unit.

EMU European Monetary Union.

FRED Federal Reserve Economic Data.

GARCH Generalized AutoRegressive Conditional Heteroskedasticity.

GBDT Gradient-Boosted Decision Trees.

GBR Gradient-Boosted Random.

GDP Gross Domestic Product.

GFCF Gross Fixed Capital Formation.

GIPS Greece, Italy, Portugal and Spain.

GJR Glosten-Jagannathan-Runkle.

GOSS Gradient-based One-Side Sampling.

IMF International Monetary Fund.

KNN K-Nearest Neighbors.

LightGBM Light Gradient Boosting Machine.

LIME Local Interpretable Model agnostic Explanation.

LR Linear Regression.

LSTM Long Short-Term Memory.

LSVR Lagrangian Support Vector Regression.

MAE Mean Absolute Error.

MART Multiple Additive Regression Trees.

ML Machine Learning.

MLP Multilayer Perceptron.

MSE Mean Squared Error.

NLP Natural Language Processing.

PACF Partial Autocorrelation Function.

ReLU Rectified Linear Unit.

RF Random Forest.

SC Sine-Cosine.

SHAP Shapley Additive Explanations.

SVM Support Vector Machine.

Tanh Hyperbolic Tangent.

VAR Vector Autoregression.

VIX Volatility Index.

WandB Weights and Biases.

WoS Web of Science.

XAI Explainable Artificial Intelligence.

XGBoost Extreme Gradient Boosting.

1 Background and Introduction

This paper aims to assess the potential of artificial intelligence in predicting sovereign bond spreads. The models are evaluated across two dimensions:

1. The accuracy in predicting the spread
2. The explainability of the models

Additionally, the paper aims to discuss the factors influencing sovereign bond spreads, and the degree to which the models align in identifying the most important determinants.

A significant amount of research on sovereign bond spreads has been done over the last 15 years due to various economic events such as the financial crisis of 2007-2008, the European debt crisis, and the COVID-19 recession. Rapid increases in bond spreads can lead to situations where governments are unable to issue new debt at sustainable rates, causing a downward spiral toward default. The World Bank's Global Economic Prospects report of 2023 highlighted the potential risk of tight global financial conditions that are raising borrowing costs across countries (World Bank, 2023). Similarly, the International Monetary Fund (IMF)'s World Economic Outlook report emphasized the risk of the combination of increased debt-to-GDP ratios and higher borrowing costs (International Monetary Fund, 2023). Therefore, understanding the factors that drive changes in the spreads is crucial. If bond spreads move in a predictable manner, it allows policymakers to take action early and avoid major economic consequences such as defaults.

It has been seen that wide bond spreads in the European Monetary Union (EMU) can undermine the effectiveness of economic policies and make it difficult to control the economy as a whole. Extreme bond spreads like those experienced during the European debt crisis, can result in a loss of trust from both investors and citizens toward the EMU. The recent economic instability in 2022-2023 has only intensified the uncertainty, further emphasizing the need to have a clear understanding of the factors that drive changes in bond spreads. As noted in the ECB's financial stability report, the recent increase in interest rates poses an economic risk in Europe, given its potential to widen bond spreads and raise the borrowing costs of indebted sovereigns (European Central Bank, 2022).

Greece, Italy, Portugal and Spain (GIPS) are of particular interest when it comes to predicting sovereign bond spreads. These countries were at the epicenter of the European debt crisis and have faced significant economic challenges, including high levels of public debt, fiscal deficits, and in some cases, slow economic growth. The economic instability of these nations has led to higher bond spreads, reflecting the increased perceived risk by investors. This makes their sovereign bond markets an interesting source of data for understanding the factors influencing bond spreads. Predictive models applied to these countries can provide valuable insights for policymakers, economists, and investors looking to better understand the dynamics of sovereign debt markets.

In order to make the best decisions possible, policymakers require access to models that are both highly accurate in their predictions and also able to provide clear explanations for the model outputs. Explainable Artificial Intelligence (XAI) seeks to bridge the gap between the predictive power of AI models and their ability to provide meaningful explanations of their outputs. Light Gradient Boosting Machine (LightGBM) and Artificial Neural Network (ANN) are both suitable models for predicting sovereign bond spreads using AI. These models have the ability to handle many features and capture nonlinear relationships between variables, making them appropriate for analyzing the interactions between economic factors that influence bond spreads. This paper applies both models and uses the Shapley Additive Explanations (SHAP) framework to provide explainability.

To evaluate the use of Machine Learning (ML) paired with XAI, it is important to have appropriate benchmarks. Various versions of autoregressive models have demonstrated strong performance in predicting sovereign bond spreads. This study applies Autoregressive (AR), Autoregressive Integrated Moving Average (ARIMA), and Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) models as benchmarks. Given that ARIMAX includes exogenous variables, it also offers a basis for comparison when it comes to explainability.

The paper proceeds as follows: Section 2 reviews relevant literature, Section 3 details the data and methodology, Section 4 presents the results, Section 5 discusses the findings and concludes, and Section 6 discusses limitations and areas for further research.

2 Literature Review

The aim of the literature review is to give a comprehensive background on the topic of predicting sovereign bond spreads. Firstly, it provides context for the work underlying this paper by highlighting significant publications and commonly used methods in the field. Furthermore, an examination of the factors affecting sovereign bond spreads is included, as well as an analysis of the use of XAI in finance. Finally, the contribution of our study to the existing literature is discussed.

2.1 Drivers of Sovereign Bond Spreads

To evaluate the existing literature in the area of prediction of sovereign bond spreads (Section 2.1 to Section 2.2), 20 papers were selected for an in-depth review. The selection was done by combining bibliometric methods and manual reviews. A detailed description of the paper selection process can be found in Appendix A. The main findings are shown in Table 1.

Country-specific macroeconomic and global markets effects
Sovereign bond spreads are fundamentally influenced by two main factors: local macroeconomic conditions and global market factors. Local macroeconomic conditions are typically used to assess intrinsic credit risk, while global market indicators proxy investor sentiment and global credit costs. The importance of local macroeconomic conditions in determining sovereign bond spreads has been consistently demonstrated in studies by Maltritz (2012), Maltritz and Molchanov (2013), Comelli (2012), Jostova (2006), De Haan et al. (2014), and Audzeyeva and Fuertes (2018). Meanwhile, Comelli (2012), Arora and Cerisola (2001), Beber et al. (2009), De Haan et al. (2014), Brooks et al. (2015), and Audzeyeva and Fuertes (2018) have conflicting results as to whether global investor sentiment and global credit costs significantly affect sovereign bond spreads.

Local macroeconomic conditions are shown to significantly affect sovereign bond spreads by Maltritz and Molchanov (2013). Furthermore, Nair (2019), Audzeyeva and Fuertes (2018), Jostova (2006), and Maltritz (2012) have demonstrated that these fundamentals provide additional predictive power. Jostova (2006) and Audzeyeva and Fuertes (2018) specifically found that local macroeconomic fundamentals improve in-sample prediction. Jostova (2006) also found that using an active trading strategy based on out-of-sample forecasting with local macroeconomic fundamentals generated twice the returns of a buy-and-hold strategy. Finally, Audzeyeva and Fuertes (2018), De Haan et al. (2014), and Beber et al. (2009) demonstrated that during times of crisis, investors tend to pay closer attention to fundamentals. In particular, Beber et al. (2009) found that liquidity becomes more important in determining sovereign bond spreads during times of crisis, while credit quality is the primary driver in less volatile times.

Generally, there are conflicting results as to whether global market sentiments and global capital costs are significantly affecting sovereign bond spreads, proxied with US market conditions and US interest rates. Maltritz (2012) and Maltritz and Molchanov (2013) find global investor sentiment to be of significance, proxied by the S&P 500 and BBB-rated US corporate bond spreads to US treasuries. However, both find the US interest rate to carry little significance. Brooks et al. (2015) found that investor sentiment towards a region as a whole can influence sovereign bond spreads. Their findings indicated that countries within the same region are perceived as equal to one another, regardless of differing macroeconomic fundamentals. US market conditions and US interest rates have also been used in studies by Brooks et al. (2015), Comelli (2012), Arora and Cerisola (2001), and Audzeyeva and Fuertes (2018).

Arora and Cerisola (2001) specifically investigated the impact of changes in US monetary policy on sovereign spreads in emerging markets. They highlighted a theoretical framework suggesting that when the US federal funds rate increased, the riskier emerging market sovereign bond yields would increase by more than the US sovereign bond yield. According to their framework, the increased yield for the riskier bonds compensates for the increased risk in holding the bond. The study argues that the increase in risk could be a result of higher debt burdens for emerging market economies. The increased risk can also cause investors to shift towards safer grounds (known as Flight-to-Safety).

Media and news effects

In recent years, there has been an increasing interest in understanding how the media can impact human decision-making. Advancements in Natural Language Processing (NLP) and ML have led to more sophisticated methods of analyzing these effects. This has resulted in an increased focus on how news sentiment can provide information for financial prediction. The power of automated Twitter accounts, or "bots," to influence the stock market and public opinion highlights the importance of media and news effects (Fan et al., 2020).

Several studies have investigated the predictive power of news sentiment in predicting sovereign bond spreads. While some studies, such as those by Erlwein-Sayer (2018) and Tobback et al. (2018), have shown that news sentiment can be useful for prediction, they did not consider whether news has predictive power in addition to macroeconomic and financial variables.

Consoli et al. (2021) examined the predictive power of news sentiment in the presence of other macroeconomic and financial variables. They found that an emotion indicator based on NLP improved forecasting power. Their emotion indicator distinguished between mild and intense fear. The study revealed that strong negative emotions are useful for short-term prediction, while milder emotions are useful for long-term prediction. Furthermore, they examined the spillover effects between Italy and Spain and observed that the Italian market primarily focused on domestic situations. In contrast, the Spanish market was influenced by news and events related to Italy.

When incorporating news into a prediction model, the choice of news sources is important. Milas et al. (2021) predicted the Greek sovereign bond spread based on the use of the term "Grexit" in both Twitter and traditional news. They found that news possesses predictive power in addition to fundamentals. Moreover, they observed a two-way information flow between tweets and traditional news, with a greater influence from Twitter to traditional news.

Study	Method	Market	Variable(s)	Period	Bond Maturity
Maltritz (2012)	Bayesian Model Averaging	Eurozone	Macro, Financial	1999-2009	10 years
De Haan et al. (2014)	OLS regression	Eurozone	Macro, Financial	2001-2013	10 years
Consoli et al. (2021)	Linear Regression	Eurozone	Financial, Alt.	2015-2019	10 years
Milas et al. (2021)	Bivariate system - VAR	Eurozone	Financial, Alt.	2012-2016	10 years
Ribeiro et al. (2017)	Dynamic panel data model	Eurozone	Financial, Alt.	2007-2016	10 years
Da Silva Fernandes et al. (2019)	SVR hybrid structures	Eurozone	Financial, Alt.	2000-2007	10 years
Erlwein-Sayer (2018)	ARIMAX	Eurozone	Alt.	2007-2017	3 months to 30 years
Tobback et al. (2018)	SVM, OLS regression	Eurozone	Alt.	2000-2013	10 years
Brooks et al. (2015)	ECM, GLS	Eurozone	Macro, Financial, Alt.	2001-2010	10 years
Beber et al. (2009)	Regression	Eurozone	Financial	2003-2004	3, 5, 7, and 10 years
Nair (2019)	OLS regression	Emerging	Macro, Financial, Alt.	2014-2017	Independent variable
Maltritz and Molchanov (2013)	BMA	Emerging	Macro, Financial, Alt.	1996-2010	Independent variable
Audzeyeva and Fuertes (2018)	Hierarchical predictive regressions	Emerging	Macro, Financial	2003-2018	3 to 20 years
de Oliveira and Montes (2021)	Regression, Machine Learning	Emerging	Macro, Financial	1996-2018	N/A
Jostova (2006)	Two-stage model, OLS regression	Emerging	Macro, Financial	1993-2001	Independent variable
Arora and Cerisola (2001)	ARCH based	Emerging	Macro	1994-1999	10 years
Comelli (2012)	OLS regression	Emerging	Macro, Financial, Alt.	1998-2011	2.5 - 30 years
Vaaler et al. (2005)	OLS based	Emerging	Alt.	1987-2000	Independent variable
Block and Vaaler (2004)	OLS, Ordinary logit	Emerging	Macro, Alt.	1987-1999	20 to 30 years
Bianchi et al. (2021)	Regression, ML	US	Macro	1971-2019	2 to 10 years

Table 1: The 20 papers that were identified to be most central regarding the prediction of sovereign bond spreads. These papers were selected for an in-depth review. Alternative variables refer to non-macroeconomic and non-financial variables used in the literature, for example, news sentiment. Bond maturity as an independent variable refers to studies that made predictions for different maturity periods, considering maturity as a factor in their prediction model.

2.2 Overview of Variables Applied in Prediction of Sovereign Bond Spreads

Financial and macroeconomic variables

After reviewing the 20 papers related to the prediction of sovereign bond spreads, an examination of the frequency of variables used was conducted. The most commonly used variables are shown in Table 2. They will now be presented, starting with the ones most commonly employed. It is important to underscore that the papers by Maltritz (2012) and Maltritz and Molchanov (2013) are of particular importance as they conducted the most comprehensive analyses of the significance of variables used in predicting bond spreads.

More than 3 uses	Other variables of interest
Fiscal balance to GDP	S&P500 and other market sentiment indicators
Real GDP growth	Money supply
CPI	Currency and reserve variables
Trade balance	Terms of trade growth
Total gov. debt to GDP	
Recent default	

Table 2: Macroeconomic and financial variables applied in the 20 papers that are used in more than 3 papers or found to be of interest.

The variable most commonly used in the reviewed papers is the fiscal balance as a percentage of Gross Domestic Product (GDP). It serves as an indicator of a government's ability to meet its financial obligations. Fiscal balance is included in the models of several researchers such as Maltritz (2012), Brooks et al. (2015), Nair (2019), Maltritz and Molchanov (2013), Arora and Cerisola (2001), Comelli (2012), and Vaaler et al. (2005). However, the significance of the variable varies across the papers.

Maltritz (2012) found a 100% chance of including fiscal balance to GDP, both with and without lag, suggesting its potential forecasting power. On the other hand, Maltritz and Molchanov (2013) reported only a 2% chance of including fiscal balance to GDP, indicating conflicting results. Maltritz (2012) found that an increase in fiscal balance leads to a reduction in the credit spread. This relationship can be explained by the fact that a rise in budget balance allows the government to generate a surplus, enabling them to effectively manage and service their debt. Despite varying outcomes, fiscal balance to GDP remains the most frequently used variable in the reviewed papers, implying its perceived importance.

Real GDP growth is a frequently used variable in the covered papers, serving as an indicator of a country's overall economic growth. It can be related to a government's ability to service debt, and a real increase in GDP can signal lower credit risk. Studies by Maltritz (2012), De Haan et al. (2014), Nair (2019), Maltritz and Molchanov (2013), Jostova (2006), Comelli (2012), and Vaaler et al. (2005) used real GDP growth in their models. Despite its common use, real GDP growth is not typically found to be a statistically significant determinant of sovereign bond spreads in the papers covered. Neither Maltritz (2012) nor Maltritz and Molchanov (2013) found real GDP growth to be significant in this regard. Nonetheless, real GDP growth remains the second most frequently employed variable.

Inflation, measured by the Consumer Price Index (CPI), is the third most widely used variable in the covered papers. Maltritz (2012), De Haan et al. (2014), Brooks et al. (2015), Nair (2019), Maltritz and Molchanov (2013), de Oliveira and Montes (2021), Jostova

(2006), Comelli (2012), and Vaaler et al. (2005) included CPI in their models. While Maltritz (2012) and Maltritz and Molchanov (2013) found inflation to be insignificant in predicting sovereign bond spreads, De Haan et al. (2014), Nair (2019), Jostova (2006), Comelli (2012), and Vaaler et al. (2005) observed it to be significant.

The variable trade balance, expressed as exports minus imports over GDP is used in studies by Maltritz (2012), Maltritz and Molchanov (2013), Audzeyeva and Fuertes (2018), and de Oliveira and Montes (2021) as an indicator of a country's ability to generate funds for debt servicing. Studies by Maltritz and Molchanov (2013) and Audzeyeva and Fuertes (2018) have found trade balance to be insignificant in predicting sovereign bond spreads. However, Maltritz (2012) found trade balance to have a positive effect on credit spread and a high probability of inclusion in the model. Additionally, Audzeyeva and Fuertes (2018) indicated that higher volatility in the trade balance, measured by its standard deviation over the most recent 6-month period, significantly improved the out-of-sample prediction. They found that increased volatility signals uncertainty in a country's ability to generate funds, which could lead to a higher credit spread.

The variable measuring the total government debt to GDP has been studied by various authors, such as Maltritz (2012), De Haan et al. (2014), and Maltritz and Molchanov (2013). A higher total debt to GDP ratio indicates an increased credit risk, leading to higher credit spreads as more debt needs to be serviced. While Maltritz (2012) suggested a low probability of inclusion in the model, the results from De Haan et al. (2014) were inconclusive. Maltritz and Molchanov (2013), on the other hand, found a high probability of inclusion in the model, with a positive relationship to the credit spread.

Another variable of interest is default history, which has been analyzed by Nair (2019), Maltritz and Molchanov (2013), and Vaaler et al. (2005). This variable indicates whether a country has recently defaulted, potentially signaling increased default risk. While Maltritz and Molchanov (2013) found a high probability of inclusion in the model, Nair (2019) did not find any significance. Vaaler et al. (2005) did not have a clear conclusion on the matter, but Maltritz and Molchanov (2013) confirmed the expected positive sign of the variable.

There are also other variables of interest, such as the S&P 500 and other market sentiment indicators, money supply, currency and reserves variables, and the terms of trade growth index. Although these variables are not as prominent, they can provide additional insight into the factors affecting credit spreads.

Alternative variables
Several approaches exist in the literature for incorporating news into prediction models. One recent study by Consoli et al. (2021) suggested that the use of an emotion-based news indicator can improve predictive power, with their focus exclusively on negative emotions. Milas et al. (2021) have found Twitter to be a superior news channel for prediction compared to traditional news sources.

2.3 Overview of Methods Used in Prediction of Sovereign Bond Spreads

Econometric Models
Many papers in the literature use regression models to predict sovereign bond spreads. One notable study by Comelli (2012) assesses the forecasting ability of three regression models using a sample data set of 29 emerging market economies. The models include a single regression model, a rolling regression model, and a gradually decreasing rolling regression

model. The results show that the third model performed the best in predicting the correct directional movement of monthly changes. Furthermore, Jostova (2006) applied a two-stage regression model, where out-of-sample testing showed that the model outperformed a buy-and-hold strategy.

Other studies in the literature have applied Generalized AutoRegressive Conditional Heteroskedasticity (GARCH)- and ARIMA-based econometric models. Ribeiro et al. (2017) use a Panel-Glosten-Jagannathan-Runkle (GJR)-GARCH-M model that includes the leverage effect. Furthermore, Erlwein-Sayer (2018) was successful in predicting bond spreads using the ARIMA and ARIMAX models. Lastly, Milas et al. (2021) applied a Vector Autoregression (VAR) model for the same purpose.

Lastly, notable studies by Maltritz (2012) and Maltritz and Molchanov (2013) used Bayesian Model Averaging (BMA) to identify the determinants of bond spreads. Their reasoning for using BMA stems from an observation that studies of sovereign bond spreads tend to have different results regarding the importance of explanatory variables. They attribute this variability to the presence of model uncertainty and potential non-linear relationships. Due to the challenges faced by traditional methods in capturing non-linear relationships, they employed BMA.

AI Models

The behavior of long-term government bond spreads is highly unpredictable over time, possibly involving non-linear relationships. This poses a challenge for traditional models, as noted by Da Silva Fernandes et al. (2019). To address these issues, machine learning has been used for predicting sovereign bond spreads in the eurozone area. In their study, Da Silva Fernandes et al. (2019) used a Lagrangian Support Vector Regression (LSVR) method that incorporates a heuristic to weigh a pool of potential inputs, such as moving averages, neural networks, and the nearest neighbor algorithm. They found that the Sine-Cosine (SC) heuristic performs better in terms of statistical accuracy than other heuristics.

Similarly, de Oliveira and Montes (2021) used machine learning to predict credit ratings and compared the performance of K-Nearest Neighbors (KNN), Gradient-Boosted Random (GBR) trees, and multi-layer perceptron methods. They concluded that the multi-layer perceptron is the most reliable machine learning method. It is important to note that de Oliveira and Montes (2021) focused on predicting credit ratings and not spreads. Nonetheless, their findings are relevant as Nair (2019) established the significance of credit ratings in predicting yield spreads. Additionally, de Oliveira and Montes (2021) discussed a trade-off between using machine learning with high predictive power and regression models with high explanatory power. The multi-layer perceptron, which was found to be the most reliable by de Oliveira and Montes (2021), had limited explanatory power. Bianchi et al. (2021) found extreme trees and ANNs to be favorable when comparing ML methods to predict US treasury returns.

When looking at the broader field of finance beyond the initially selected 20 papers, tree ensemble methods such as Extreme Gradient Boosting (XGBoost) and LightGBM have been used with success. Qian et al. (2022) used XGBoost and LightGBM to predict financial distress for Chinese companies, and found the methods to outperform methods such as Linear Regression (LR), ANN, Support Vector Machine (SVM), and Random Forest (RF). In a cryptocurrency price prediction study, Sun et al. (2020) found LightGBM to be more robust than SVM and RF. LightGBM also has the advantage of faster computation speeds compared to other boosting methods, which is an advantage when dealing with large datasets, as seen by e.g. Bentéjac et al. (2021).

2.4 Explainable Artificial Intelligence

The field of XAI has gained interest in recent years, but the need for interpretability in prediction models dates back earlier. Research on interpretable prediction models gained momentum in the 1990s (Wick and Thompson, 1992; Swartout and Moore, 1993) and has since led to the development of frameworks and techniques to balance accuracy and explainability in machine learning prediction models.

The XAI techniques can be broadly categorized by scope, stage, and model as outlined by Kamath and Liu (2021). Scope refers to the level of interpretation, which can be either local or global. Stage refers to the point of interpretation, pre-model, post-model, or intrinsic. The model category denotes whether the technique is specific to a particular model or can be applied to any model, known as specific or agnostic. There are also alternative ways of categorizing XAI techniques, such as the taxonomies of Arrieta et al. (2020) and Minh et al. (2022).

The SHAP framework is a widely-used tool that falls under the categories of post-model and model-agnostic technique that can provide both local and global explanations. The concept of Shapley values was first introduced to game theory by Shapley et al. (1953) to measure the marginal contribution of each player to the overall value. This offers a way to distribute credit for a model's prediction among its input features, providing the relative importance of each feature in determining the final prediction. The SHAP framework by Lundberg and Lee (2017b) applies the concept of Shapley values to offer insight into the factors that contribute to a prediction in machine learning.

Two alternatives to the SHAP framework are the Local Interpretable Model agnostic Explanation (LIME) and Explain Like I'm 5 (ELI5) frameworks. LIME was proposed by Ribeiro et al. (2016) and learns an interpretable model around the prediction locally. ELI5 aims to provide quick and easy-to-understand explanations of a model's predictions. Out of the three methods, SHAP was found to be the desired one by Vij and Nanjundan (2022). SHAP has been applied widely in financial literature and has proven to be consistent with human intuition (Demajo et al., 2020).

The SHAP framework provides multiple explainers, of them are TreeExplainer (Lundberg et al., 2020) and KernelExplainer two of the most notable (Vij and Nanjundan, 2022). TreeExplainer specializes in generating Shapley values of tree-based models while the KernelExplainer is an explainer that is generic in nature and can be employed to interpret any type of model. The SHAP framework also offers a general explainer, which is the primary explainer interface for the SHAP library. For the general explainer, one can set the algorithm used for generating Shapley values to `permutation`, `partition`, `tree`, `linear`, or `auto`. The default value is `auto`, where the general explainer attempts to make the best choice for what algorithm to use on the given model.

KernelSHAP has become widely used in financial literature. Mokhtari et al. (2019) applied KernelSHAP in explaining the classification of various prediction methods used in financial time series such as KNN, SVM, RF, XGBoost, and Long Short-Term Memory (LSTM). However, due to its high computational complexity, KernelSHAP is relatively inefficient, as noted by Liu et al. (2022). Therefore, TreeSHAP has become a popular tool for calculating SHAP values for tree-structured ML models. For example, Bussmann et al. (2021) used TreeSHAP to explain an XGBoost model that predicts default risk.

KernelSHAP assumes independence among features (Aas et al., 2021). Therefore, Aas et al. (2021) argue that Shapley values can produce misleading explanations when features are correlated. They extend KernelSHAP to handle dependent features and show that it gives estimations to true SHAP values that are more reliable.

2.5 Contribution to the Literature

This research aims to add to the existing body of literature on the topic of predicting sovereign bond spreads by exploring the potential of ML and XAI. To the best of the authors' understanding, there has been no application of XAI techniques for this particular use case, and this study seeks to fill that gap. Furthermore, there has generally been limited research on the topic of AI to predict sovereign bond spreads. By examining the feasibility and accuracy of using AI methods for predicting sovereign bond spreads, this paper contributes new insights into the topic of sovereign bond spread prediction.

The field of AI in finance has seen significant growth in recent years, with numerous studies exploring the potential of AI in financial forecasting and decision-making. However, a common limitation of these studies is that they often only compare AI models to simple benchmark methods such as linear regression, which can be insufficient in accurately evaluating the usefulness of AI in finance. This study seeks to address this limitation by comprehensively benchmarking AI to econometric models in the domain of sovereign bond spread prediction and thus giving a complete picture of the potential of AI. The comparison is carried out on both accuracy and explainability.

Lastly, the study contributes to the existing literature on the determinants of sovereign bond spreads. The research examines the factors identified by the models and discusses their alignment or divergence. This analysis provides insights into the key factors affecting sovereign bond spreads.

3 Data and Methodology

First, this section gives an overview of how the data was prepared. Additionally, the ML models, econometric benchmarks, and explainability framework used will be introduced. Furthermore, the hyperparameters for each model and how they were tuned will be described. Finally, the performance metrics used for evaluating the results will be presented.

3.1 Data Collection and Cleaning

When preparing data, several important steps had to be taken. Firstly, the process involved collecting a set of candidate variables based on the literature discussed in Section 2. Secondly, certain candidate variables were removed in the data-cleaning process, resulting in a dataset of 51 explanatory variables. Thirdly, 30 of the 51 explanatory variables were chosen through feature selection. Finally, the data was sampled for appropriate implementation in the ML models and gathered in a pandas DataFrame (McKinney et al., 2010).

Data Collection
We identified candidate variables by considering all the economic and financial variables used in the 20 papers identified in our literature review (Section 2). The data was collected from three different sources, namely Federal Reserve Economic Data (FRED) (Federal Reserve Bank of St. Louis, 2023), the Refinitiv Eikon database (Refinitiv, 2023), and The Heritage Foundation (The Heritage Foundation, 2023). The main source of data used in this research came from FRED, an accessible online database that provides a wide range of economic data. The Eikon database was used as a supplementary source in cases where data was either unavailable or of unsatisfactory data quality on FRED. Furthermore, the Heritage database was used as a source of freedom indicators and political data. The process of extracting data from the sources is thoroughly explained in Appendix B.

Data Frequency
The collected data encompasses a range of frequencies, from real-time financial data to annual political and macroeconomic data. However, the majority of the data was available on a quarterly basis. We determined that a quarterly frequency would be advantageous to be able to use most of the available data. Using a quarterly frequency balanced the generation of too much synthetic data from disaggregation and the limitation of too few data points from aggregation. This approach contrasts with the conventional method of aggregating all the data to the least frequent sampling frequency, which, in this case, was annual. Prior to applying the models, the data was organized into a common dataset, ensuring fair comparison among the models. To achieve this, the higher-frequency data was aggregated while the annual-frequency data was disaggregated.

The majority of the annual data applied in the analysis consisted of political indicators, such as corruption levels, property rights, and investment freedom. Since general trends characterize these indicators, we used cubic spline interpolation to disaggregate the data into quarterly frequency. Cubic spline interpolation was introduced by Schoenberg (1946) and the idea is to find a function that fits the given data points and has continuous first and second derivatives. This method is advantageous when working with general trends in the data. The interpolation was performed in Python (Van Rossum and Drake, 2009) using the SciPy CubicSpline function (SciPy community, 2021). We applied endpoint interpolation to aggregate higher-frequency data into quarterly data. This allowed for a complete and consistent quarterly frequency dataset.

Dependent Variables

The dependent variables were derived from the 10-year sovereign bond yields obtained from Refinitiv Eikon. In the European market, it is conventional to use the German yield as the benchmark when assessing sovereign bond spreads. To calculate each spread, we subtracted the 10-year yield of the German sovereign bond from the yield of the 10-year sovereign bond for the relevant country. The resulting four dependent variables are shown in Table 3.

Dependent variables
Greece-Germany 10-Year Sovereign Bond Spread
Italy-Germany 10-Year Sovereign Bond Spread
Portugal-Germany 10-Year Sovereign Bond Spread
Spain-Germany 10-Year Sovereign Bond Spread

Table 3: The four dependent variables applied. All of them were considered separate observations under the same general AI model, where the feature data in each sample corresponded to the respective country in the target variable (see Section 3.3).

Explanatory Variables

Once the data frequency was determined, a manual assessment was conducted to evaluate the relevance of each variable based on its availability within the 1999-2020 time period. The year 1999 marks the final stage of the commencement of the EMU (European Central Bank, 2023). This ensures that the spreads were not affected by exchange rate risk. The year 2020 marks the start of the COVID-19 pandemic in Europe, which led to asset purchase programs, substantial shifts in yield curves, and other extreme changes in economic variables over a short time period. This poses challenges for prediction using conventional methods, especially with a quarterly data frequency. Hence, the timeframe from 1999 to 2020 was determined to be an appropriate duration. Variables that had missing values persisting for extended periods within the timeframe, and were not available from other sources, were excluded from further analysis. Tables 20, 21, and 22 in Appendix B present the candidate variables obtained from FRED, Refinitiv Eikon, and Heritage, respectively.

To ensure that general models could be trained to compare data across countries, relevant ratios were calculated for the country-specific variables. An example is that Gross Fixed Capital Formation (GFCF) as a percentage of GDP was used instead of absolute GFCF. Furthermore, the country-specific variables were differenced by observations for Germany. This approach allowed for meaningful comparisons between countries. To ensure the preservation of information during the differencing process, an additional category of variables named ”Eurozone Conditions” was introduced, incorporating the absolute value of the German observations. These variables were included alongside the Global Conditions and Country Specific variables. Lastly, binary dummy variables for the countries were included. The significance of the dummy variables is interesting for understanding whether there exist country-specific factors that are not captured by the other variables.

This resulted in the 51 explanatory variables shown in Table 4.

3.2 Feature Selection

Performing feature selection prior to training a ML model leads to noise reduction in the input. This noise reduction enhances the performance of the ML models by minimizing the chances of identifying non-causal relationships between the target and explanatory

Explanatory variables	**Type**
Constant GDP per capita	Country Specific
Constant GDP growth	Country Specific
GFCF(% of GDP)	Country Specific
Import of goods and services(% of GDP)	Country Specific
Export of goods and services(% of GDP)	Country Specific
Current account balance (% of GDP)	Country Specific
CPI all items total	Country Specific
CPI all items non-food non-energy	Country Specific
Return on regional stock index	Country Specific
Unemployment	Country Specific
Government debt (%of GDP)	Country Specific
Property rights	Country Specific
Government integrity	Country Specific
Tax burden	Country Specific
Government spending	Country Specific
Business freedom	Country Specific
Monetary freedom	Country Specific
Trade freedom	Country Specific
Investment freedom	Country Specific
Financial freedom	Country Specific
S&P500 return	Global Conditions
FED interest rate	Global Conditions
Bond yield US treasury 1y mat.	Global Conditions
US BBB-rated corp. bond over treasuries	Global Conditions
Prime lending rate	Global Conditions
Crude brent europe	Global Conditions
10y US gov minus 90-day US bill	Global Conditions
VIX	Global Conditions
Local interbank rate	Eurozone Conditions
EU policy rate	Eurozone Conditions
DE constant GDP per capita	Eurozone Conditions
DE constant GDP growth	Eurozone Conditions
DE GFCF(% of GDP)	Eurozone Conditions
DE current account balance (% of GDP)	Eurozone Conditions
DE import of goods and services(% of GDP)	Eurozone Conditions
DE export of goods and services(% of GDP)	Eurozone Conditions
DE CPI all items total	Eurozone Conditions
DE CPI all items non-food non-energy	Eurozone Conditions
DE return on regional stock index	Eurozone Conditions
DE unemployment	Eurozone Conditions
DE government debt (%of GDP)	Eurozone Conditions
DE property rights	Eurozone Conditions
DE government integrity	Eurozone Conditions
DE tax burden	Eurozone Conditions
DE government spending	Eurozone Conditions
DE business freedom	Eurozone Conditions
DE monetary freedom	Eurozone Conditions
DE trade freedom	Eurozone Conditions
DE investment freedom	Eurozone Conditions
DE financial freedom	Eurozone Conditions
Greece	Binary Dummy
Italy	Binary Dummy
Portugal	Binary Dummy
Spain	Binary Dummy

Table 4: The 51 explanatory variables tested. The country-specific variables are differenced by the observation from Germany, and the German observation is included in the dataset as a European Conditions variable labeled with ”DE” (Deutschland). After performing feature selection, the number of features was reduced to 30.

variables (Gareth et al., 2013). In this scenario, where the number of features is notably high compared to the dataset's row count, it becomes essential to reduce the number of features. This step is important to address overfitting issues effectively and ensure the development of a reliable model.

We used a recursive feature elimination method by scikit-learn (Pedregosa et al., 2011) to perform feature selection. Recursive feature elimination first fits a given model on all the features and subsequently removes the least important feature. This procedure is repeated recursively until a given number of features remain. The objective of feature selection is to decrease the number of features while ensuring the retention of features that could potentially be significant. The use of this method provided us with the flexibility to manually select the desired number of features. This is advantageous because automated methods for determining the number of features can exclude important features when non-linear relationships exist.

The model used for feature selection was a default tuned LightGBM model. This may introduce the risk of a bias in favor of the LightGBM model in the results. We still found it preferable to use this method for feature selection as it would allow us to identify important features through non-linear relationships. Table 5 contains the selected top 30 features that were used for further analysis.

Variable name	**Explanation**
GDP_CAPITA	Constant GDP per capita (differential)
GDP_GROWTH	GDP growth rate to the same period the previous year (differential)
GFCF	Gross Fixed Capital Formation % of GDP (differential)
EXPORT	Export of goods and services % of GDP (differential)
CPI_TOT	Consumer Price Index: all items (differential)
CPI_NON_FOOD	Consumer Price Index: all items non-food non-energy (differential)
LOCAL_STOCK_INDEX	Return on the stock index in the relevant country
UNEMPLOYMENT	Harmonized unemployment rate (differential)
GOVDEBT	General government gross debt % of GDP (differential)
GOVINT	Government integrity (differential)
GOVSPE	Government spending (differential)
MONFRE	Monetary freedom (differential)
INVFRE	Investment freedom (differential)
FINFRE	Financial freedom (differential)
DFF	US federal funds effective rate
BBB_yield_US	BBB US corporate index effective yield
SP500_RETURN	S&P 500 return
VIX	CBOE Volatility Index
ECB_INTERBANK_RATE	3-Month interbank rate
DE_GDP_CAPITA	Germany: constant GDP per capita
DE_IMPORT	Germany: export of goods and services % of GDP
DE_CPI_TOT	Germany: Consumer Price Index all items
DE_UNEMPLOYMENT	Germany: harmonized unemployment rate
DE_GOVDEBT	Germany: general government gross debt % of GDP
DE_PRORIG	Germany: property rights
DE_GOVSPE	Germany: government spending
10YT_SPREAD_LAG1	Target variable lagged one quarter
10YT_SPREAD_LAG2	Target variable lagged two quarters
10YT_SPREAD_LAG3	Target variable lagged three quarters
10YT_SPREAD_LAG4	Target variable lagged four quarters

Table 5: The 30 features selected. All exogenous variables use the first lag. The selection was done using recursive feature elimination.

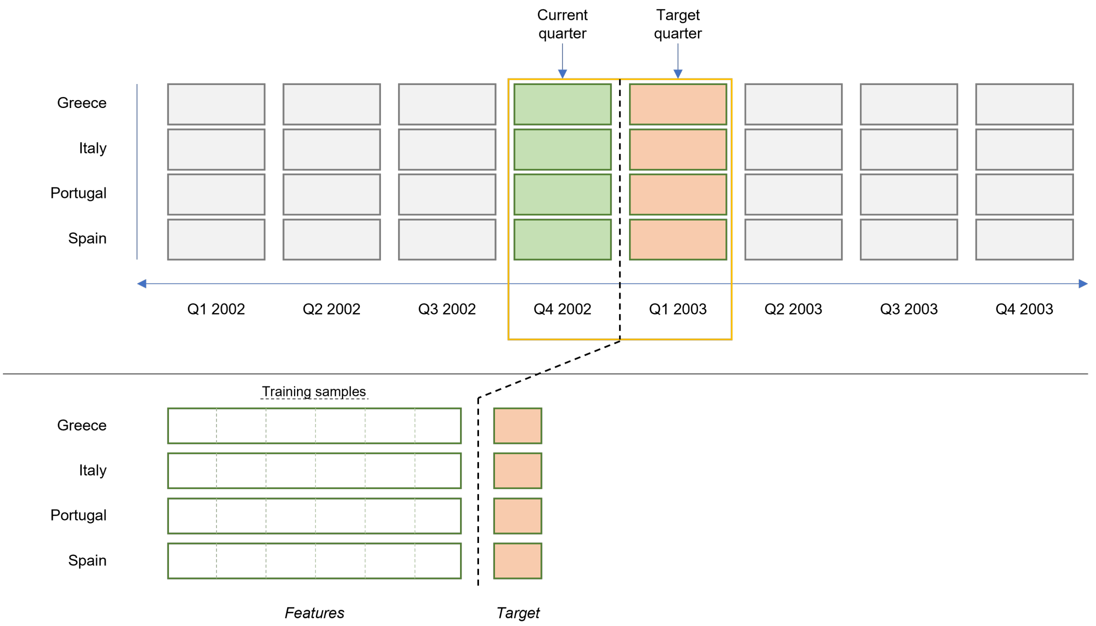


Figure 1: The sample creation process for AI models involved eliminating the time dimension from the samples, resulting in feature observations and a corresponding target.

3.3 Sample and Target Generation

To create feature-target pairs for the ML models, a rolling window sampling method, as seen in Figure 1, was employed. Therefore, each sample consists of a target for a specific country, and the explanatory variables related to that country, as well as other general economic indicators. The exogenous variables are all lagged one-quarter behind the dependent variable.

Sampling the data in this way can lead to bias if time-dependent effects are not taken care of. To avoid this, the data were transformed into general ratios, as described in Section 3.1. This also made it possible to treat the spreads from different countries as different observations under the same general model. Four lags of the target value were included in the dataset, as the literature has shown yield curve data to be important. This is a commonly used lag length in the literature when dealing with quarterly data (see Section 2.3). A four-quarter lag length means that our initial target variable corresponds to Q1 2000, while the dataset begins in 1999.

Finally, the data were divided into three sets: training, validation, and testing. The training set was employed to train the models, the validation set was used to tune the hyperparameters, and the test set was reserved to evaluate the performance of the models. Testing will be conducted on the final 12 quarters of the dataset, namely from Q1 2017 to Q4 2019, accounting for approximately 15% of the total dataset. Accordingly, the preceding 12 quarters from 2014 to 2017 were chosen as the validation set, while the remaining data covering 1999 to 2014 were allocated as the training set.

3.4 Data Analysis

Table 8 shows an overview of the descriptive statistics for the selected features. From the table, we see large variances between the features with respect to maximum and minimum values. LightGBM is robust to data with large values, as it is a tree-based model (Gareth et al., 2013, Chapter 8). In contrast, ANNs tend to yield better results when the input data features have an even distribution, without significant variations (Goodfellow et al., 2016, p. 299). In order to address the scale differences for the ANN, we used scikit-learn's Quantile Transformer (Pedregosa et al., 2011).

As we will discuss in Section 3.7, we evaluate the performance of the regressors by how well they predict upward or downward movements, in addition to using conventional regression measures. For the classification task, it is important to explore the balance of the data. In Table 6 we see that the total dataset contains approximately equally many upward and downward movements. Hence, there is no need to modify the dataset with techniques like oversampling. However, the test sets only have 37.5% upward movements on average. Therefore, we introduce a measure to look for potential bias in the models in Section 3.7.

Country	**Total set**	**Test set**
Greece	0.500	0.333
Italy	0.4875	0.500
Portugal	0.425	0.250
Spain	0.4375	0.417
Average	0.4625	0.375

Table 6: The positive quarterly directional movement fraction for each country, both within the complete dataset and the test set. It is calculated by dividing the number of positive quarterly moves by the total quarterly moves.

Table 7 gives an overview of the country-specific 10Y bond spread statistics. We see that Greece has more extreme values and a larger standard deviation compared to the other countries. The country with the second highest standard deviation is Portugal, while Italy and Spain have the lowest.

Country	**Mean**	**Median**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Greece	4.7264	2.2520	6.2279	0.1180	0.3090	2.2520	7.8650	33.6720
Italy	1.2257	1.0670	1.1235	0.087	0.265	1.067	1.685	5.219
Portugal	1.9553	0.807	2.5683	-0.0670	0.2030	0.8070	2.5550	11.7410
Spain	1.0140	0.7190	1.1561	-0.0770	0.0920	0.7190	1.2800	4.9150

Table 7: Descriptive statistics of the 10Y bond spread (target variable) for each country

Feature Name	Mean	Median	Std	Min	25%	50%	75%	Max
GDP_CAPITA	-13750.1067	-14915.2054	6150.3302	-24381.1024	-17387.4586	-14915.2054	-14915.2054	-1722.2733
GDP_GROWTH	-0.5278	0.0422	3.2746	-17.5444	-1.8254	0.0422	0.0422	7.1159
GFCF	0.0305	0.4365	4.9796	-12.2884	-3.1075	0.4365	0.4365	10.7957
EXPORT	-12.0521	-12.3409	5.6109	-25.4167	-16.4365	-12.3409	-12.3409	3.3937
CPI_TOT	0.4767	0.4780	1.2787	-3.1385	-0.4845	0.4780	0.4780	4.3575
CPI_NON_FOOD	0.5245	0.5360	1.4038	-3.5589	-0.5084	0.5360	0.5360	4.0696
LOCAL_STOCK_INDEX	-0.0017	-0.0025	0.1176	-0.3758	-0.0814	-0.0025	-0.0025	0.4058
UNEMPLOYMENT	6.0515	3.7000	7.3711	-3.6000	0.0000	3.7000	11.1000	23.2000
GOVDEBT	105.0396	105.6518	38.1430	35.7261	73.1730	105.6518	130.0239	190.6740
GOVINT	-25.0489	-22.6490	11.1064	-47.6288	-34.4286	-22.6490	-16.7369	-1.9129
GOVSPE	-0.3687	-1.2326	15.5311	-42.7229	-12.1076	-1.2326	12.5297	28.3538
MONFRE	-2.4601	-2.0267	3.7737	-12.6000	-3.9435	-2.0267	-0.9000	9.6000
IINVFRE	-15.6522	-18.1842	11.5933	-41.4765	-20.9771	-18.1842	-6.3009	6.0506
FINFRE	-0.2546	0.0000	13.9079	-30.8330	-10.0000	0.0000	4.0729	30.0000
DFF	1.8422	1.3300	1.9572	0.0400	0.0900	1.3300	2.7500	6.8600
BBB_yield_US	5.3158	4.9200	1.5979	3.3000	4.0300	4.9200	6.2900	9.6700
SP500	1.1641	1.9030	7.7995	-21.8710	-3.5960	1.9030	7.8400	15.6050
VIX	19.7819	17.7400	7.7563	9.5100	13.9500	17.7400	23.9500	44.1400
ECB_INTERBANK_RATE	1.7309	1.4886	1.7559	-0.4176	0.0971	1.4886	3.3101	5.0192
DE_GDP_CAPITA	38082.8417	37797.1550	2995.3461	33345.6006	35034.0806	37797.1550	40689.2778	43357.6984
DE_IMPORT	35.8079	37.4311	4.5226	26.8194	31.0975	37.4311	39.6492	42.0527
DE_CPI_TOT	1.4521	1.4115	0.6907	-0.2018	1.0764	1.4115	1.8640	3.2359
DE_UNEMPLOYMENT	6.7173	7.0000	2.4738	2.9000	4.7000	7.0000	8.5000	11.2000
DE_GOVDEBT	68.1871	66.4493	7.4395	58.1930	62.2618	66.4493	74.4517	82.0207
DE_PRORIG	88.5417	90.0000	3.2889	79.9000	89.9930	90.0000	90.0000	90.6742
DE_GOVSPE	36.2693	36.7627	4.8108	26.6001	32.2730	36.7627	41.2237	43.6826
TARGET	2.2303	0.9950	3.7536	-0.0770	0.2518	0.9950	2.5288	33.6720

Table 8: Descriptive statistics for the selected exogenous variables and the target variable. The table displays the data after implementing the relevant ratio or differential calculations as indicated in Table 5, which explains certain negative values.

3.5 Models and Hyperparameters

This subsection provides an overview of the models used, namely ANN, LightGBM, and econometric benchmarks. Additionally, the most important hyperparameters for each model will be explained in detail. Hyperparameters can significantly influence the learning process, and finding the appropriate hyperparameters in ML models requires extensive testing. Weights and Biases (WandB), a tuning Application Programming Interface (API), was used in this process. The API automates the tuning process and provides visual representations of the results (Biewald, 2020). A comprehensive description of WandB and how it was used to test the models' hyperparameters is presented in Appendix F.

All models, including the benchmark models, used an expanding window for making out-of-sample predictions on the test set, as seen in Figure 2. This allowed us to include as much data as possible given the relatively small size of the data set. The process involves training the model using the training and validation datasets, followed by iterative prediction on the test set. After predicting each test row, which corresponds to a one-quarter-ahead prediction, that row is included in the training set and the model is retrained. This process continues iteratively until predictions are made for all the rows in the test set.

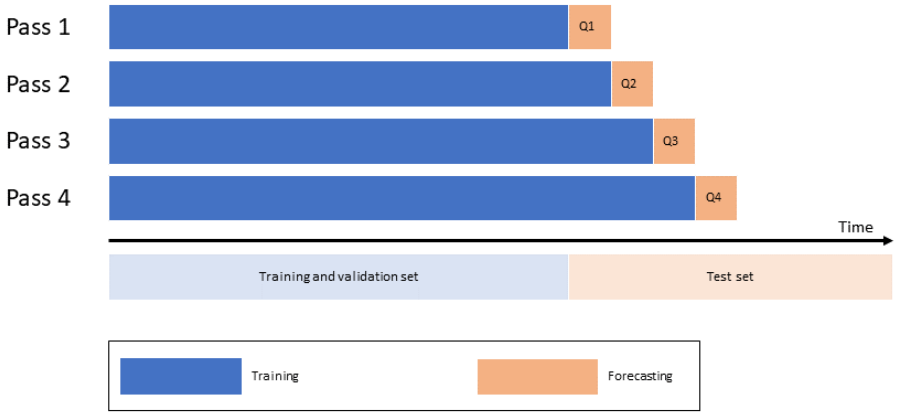


Figure 2: Illustration of expanding window method for predictions on historical data.

The AI models applied sample data from all four countries, but the hyperparameters were set for each country individually. This allowed the models to specialize for each country while still benefiting from more training data. This means that the individual models, with country-specific hyperparameters, made predictions for that particular country. Then it added four new rows (one from each country) to the training set before making the prediction for the next quarter. In contrast, the econometric models were trained individually for each country, using exclusively the data from that particular country, in addition to the global data used in all the models.

3.5.1 Artificial Neural Network

In Section 2.3, we discussed that artificial neural networks (ANNs) are commonly used and flexible models. However, we have not found any previous studies that have combined ANNs with explainable artificial intelligence (XAI) techniques for predicting sovereign bond spreads. Furthermore, as mentioned by Maltritz (2012) and Maltritz and Molchanov (2013), there may be a model uncertainty when modeling sovereign bond spreads, which may explain the varying results regarding variable significance. An ANN does not need predefined relations between the variables and thus circumvents this problem. Therefore, we aim to evaluate the effectiveness of a simple feedforward ANN model in this specific area. For implementing the ANN, we chose Keras (Chollet, 2015), a Python library built on top of TensorFlow, as it offers extensive documentation on various hyperparameters. Keras also integrates with the tuning API WandB. Creating an ANN involves making several decisions, including configuring the hyperparameters outlined in Table 9. In the following sections, we provide detailed explanations of each parameter in the table. For additional background on ANNs please refer to Appendix C.

Activation function

The activation function in a perceptron decides the manner in which the input weights are transformed into an output. The choice of activation function for the hidden layers of a neural network can affect the network's ability to learn the task at hand, whereas the activation function in the output layer determines the type of predictions the network is capable of making. Since this regression task aims to predict continuous output values without any predefined bounds, a linear activation function was used in the output layer. For the hidden layers, the Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), and Hyperbolic Tangent (Tanh) activation functions were all tested. For more discussion on activation functions see Appendix C.

Number of Hidden Layers

According to Zhang et al. (2021), the number of hidden layers is an important aspect of model design that directly influences the complexity of the model. They emphasize that a model with a single input and output layer can only achieve linear separation, limiting its ability to capture complex relationships in the data. Furthermore, the universal approximation theorem, demonstrated by Cybenko (1989) and further refined by Hornik et al. (1989), states that a neural network with a single hidden layer and a nonlinear activation function can approximate any continuous decision boundary effectively.

In terms of the risk of overfitting, a study by Hinton and Van Camp (1993) discusses how increasing the number of hidden layers can lead to overfitting. They highlight the importance of balancing model complexity and generalization ability to achieve optimal performance. As we aim to approximate any arbitrary and continuous decision boundary, while still keeping the number of hidden layers to a minimum, we test for 2 and 3 hidden layers.

Number of Neurons in Each Hidden Layer

The nodes in the hidden layers are referred to as `units` and `neurons`. Determining the optimal number of neurons in each layer involves balancing the need to capture complex connections against the risk of overfitting to the training data. Xu and Chen (2008) argued that using too few neurons leads to high training and generalization error due to underfitting, while too many neurons result in lower training error due to overfitting. Rules of thumb and research on the optimal number of neurons exist, however, trial and error remain the most common approach (Xu and Chen, 2008). Therefore, various number of neurons per hidden layer was tested as shown in Table 9.

Learning Rate

The learning rate is a crucial hyperparameter in training neural networks as it significantly impacts the performance of the model. The parameter is named `adam_lr` in Figure 15. If the learning rate is set too low, the training process may become slower and result in convergence to a local optimum, while setting it too high can lead to divergence. There are ways to assist in the search for good learning rates, such as the Cyclical Learning Rate (CLR) policy proposed by Smith (2017). The downsides to such methods are training instability, added complexity, and increased sensitivity to new hyperparameters. Therefore, we opted for using a simple uniform search between the values 0.1 and 0.001.

Regularization
Regularization techniques for ANNs are focused on preventing overfitting by controlling model complexity. They involve adding constraints, penalties, or modifications to the training process to encourage the network to generalize well on unseen data. Dropout is a form of regularization method for ANNs, that was introduced by Srivastava et al. (2014). Dropout randomly deactivates neurons along with their corresponding weights and biases during training. Practically, it means that during each training iteration, a random subset of neurons is picked and the model is trained on only those neurons. This decreases the impact of noise from the training data as the neurons have to work independently, which leads to better generalization. The dropout rate determines the proportion of neurons that will be excluded from the training on average. Srivastava et al. (2014) suggests a dropout rate of 0.5, and we tested values between 0.4 and 0.8.

Batch Size
Batch size is the number of rows used in each forward pass and backpropagation iteration. The model adjusts its weights and biases for each batch it processes. Large batch sizes can be used to optimize processing time but may converge slower. Reducing the batch size can introduce some noise, but it can enhance generalization and aid in avoiding local minima. According to Masters and Luschi (2018), it is recommended to test various batch sizes. Hence, we test batch sizes of 32 and 64, as smaller sizes tend to generalize better.

Epochs
Epochs are the number of times all of the training data is processed by the ANN. A run through all the batches thus equals 1 epoch. By setting the number of epochs too low, one runs the risk of the model not capturing all the relevant patterns in the dataset. However, by setting the number of epochs too high, one increases the risk of overfitting the model. Goodfellow et al. (2016) encourage empirically testing different epoch values, and we have done so by testing a wide range of values as seen in Table 9.

Optimizer
In ANNs, the optimizer is the algorithm or method used to adjust the weights and biases of the neural network during training. The optimizer minimizes the loss function, with the goal of making the network better at predictions. We have chosen to use Adam (Kingma and Ba, 2014). Adam's downsides mainly stem from its lack of theoretical convergence guarantees, memory requirements, and sensitivity to the initial learning rate. However, it is widely used and well established, is robust to hyperparameter tuning compared to other methods, and shows empirically efficient convergence. We did not test multiple optimizers, as it would increase the search space and complexity while having negligible effects on the results. Optimizers mainly influence the speed and efficiency of the training and have little impact on the results compared to other hyperparameters.

3.5.2 Light Gradient Boosting Machine

The LightGBM framework was selected based on several factors. Firstly, it has a great performance on tabular data and remarkable training speed and accuracy. Additionally, it is a widely used method in the literature (see Section 2.3). As the framework has a tree-based structure, there is no need for input data scaling or transformation. Furthermore, it can operate optimally without relying on assumptions about the input distribution. Lastly, just as for an ANN, it does not assume relations between features, which circumvents the

Hyperparameter	**Values**
Activation function (in hidden layers)	ReLU, ELU, Tanh
Number of hidden layers	(2, 3)
Number of units per hidden layer	(32, 64, 128, 256)
Learning rate	($U_{0.001,0.1}$)
Dropout	(0.4, 0.5, 0.6, 0.7, 0.8)
Batch size	(32, 64)
Epochs	(32, 64, 128, 256)
Optimizer	Adam

Table 9: The hyperparameter search space for the ANNs. The U denotes the search was done across a uniform distribution.

modeling uncertainty problem proposed by Maltritz (2012) and Maltritz and Molchanov (2013). For additional background and intuition, please refer to Appendix D.

The LightGBM algorithm contains more than 100 parameters that can be tuned. However, only the most influential parameters were selected for tuning. A summary of these hyperparameters, as well as the different values for each parameter that was tested, is presented in Table 10. We implemented LightGBM in Python using Microsoft's LightGBM framework (Microsoft, 2023).

Gradient Boosting Methods

LightGBM is a gradient-boosting framework that leverages the concept of progressively improving a weak learner (Kearns, 1988). By iteratively adding new weak learners that enhance the predictions made by the existing ones, the overall predictive power of the learner ensemble is improved. LightGBM, specifically, is a gradient-boosting framework that employs decision trees as weak learners (Friedman, 2001). It uses the gradient of a specified loss function to add new weak learners. Decision trees partition a dataset into subsets based on a specific feature and its potential values.

The LightGBM framework has the option to specify one of three different gradient boosting methods; Gradient-based One-Side Sampling (GOSS), Gradient-Boosted Decision Trees (GBDT), and Dropout meet multiple Additive Regression Trees (DART). GBDT is the original proposed method by Friedman (2001). DART is an extension that incorporates the regularization technique *dropout* into gradient boosting algorithm. For gradient boosting methods, this means ignoring a part of the existing trees when creating new trees. Lastly, GOSS is what makes LightGBM fast compared to other methods. This approach primarily focuses on training on the samples considered challenging, characterized by high gradients, while including only a small subset of the samples identified as easy, which have low gradients.

Setting the `boosting_type` hyperparameter to GOSS is necessary to replicate the model proposed by Ke et al. (2017) in LightGBM. In our case, using GOSS allows us to train and test a broad range of model configurations during hyperparameter tuning within a reasonable computational frame. Therefore, the boosting parameter was set to GOSS. For a more comprehensive evaluation of all three boosting parameters options, please refer to Appendix D.

Regularization

As with any other machine learning model, one of the main concerns with gradient boosting models is the possibility of overfitting. As the LightGBM framework constructs weak learners leaf-wise and not level-wise, the algorithm will converge faster but also be more prone to overfitting (Zhang and Gong, 2020). To combat this, there are five regularization techniques used in boosting models; regularization of weak learners, regularization of the loss function, constraints on the number of iterations, random sampling, and shrinkage - setting a small learning rate to control the contribution of each new learner. Below we present the hyperparameters we will tune to avoid overfitting.

Regularization of Weak Learners
In our case, the regularization of weak learners involves imposing constraints on the number of leaves and the maximum depth of each tree. The leaf nodes in the tree represent the final predicted outcomes. By limiting the number of leaves in the weak learners, we reduce the prediction diversity of each learner, making them more generalized in their predictions. Consequently, this mitigates the risk of overfitting. The hyperparameter `num_leaves` manages this characteristic. Similarly, the maximum depth of a tree determines the number of times it can partition the feature space before arriving at a conclusion. By restricting the maximum depth, we decrease the individual specialization of trees, leading to a more generalized model. The hyperparameter `max_depth` is employed to tune this aspect. Tuning these parameters involves a trade-off between model complexity and generalization. Trees with many leaves and great depth have the capacity to capture intricate relationships but are more susceptible to overfitting. Given that leaf-wise trees tend to reach deeper depths earlier, it is crucial to fine-tune these parameters to strike an optimal balance.

Regularization of the loss function
Regularization of the loss function involves the use of an extra penalty term in the loss function. The two regularization techniques most widely used for this are L1 and L2 regularization, also called lasso regression and ridge regression, respectively. L1, or lasso, adds the absolute value of the magnitude of the given coefficient as a penalty to the loss function. As the loss function is minimized, the absolute values of the coefficients are pushed down. L2, or ridge, adds the squared magnitude of the coefficient. Just like the L1 technique, this will also push the size of the coefficients down. For both methods, there is a regularization term, λ, multiplied by the penalty term. λ is the hyperparameter that adjusts the effect of the penalty terms. Setting λ equal to 0 completely removes the effect of the penalty term, while increasing λ towards 1 will progressively increase the penalty. A key difference between L1 and L2 is the fact that L1 pushes the values of the coefficients to zero, as opposed to L2. This makes L1 work great for feature selection as it discards less important features completely. For this reason, we use L2 regularization, as feature selection was already performed. The parameter used to tune the λ for L2 regularization is `lambda_l2`.

Constraint on the Number of Iterations
The number of iterations in a gradient boosting algorithm refers to the number of total learners the algorithm produces, as a new tree is created on each iteration. Intuitively, adding a new learner will make the method more specialized and able to find more complex patterns in the dataset. However, as with the other properties discussed above, there is a trade-off between complexity and generalization. Limiting the number of iterations with the hyperparameter `num_iterations` will reduce the probability of overfitting. Another way of minimizing overfitting through the number of iterations is monitoring the model's performance on the validation set and stopping the training early if there hasn't been a satisfactory increase in performance over a given number of iterations. This process is tuned with the `early_stopping_rounds` hyperparameter, which determines the number of iterations without notable improvement the algorithm runs before ending the training process.

Shrinkage

In gradient-boosting, the `learning_rate` determines how much each new tree contributes to the tree ensemble. This is also known as shrinkage. A learning rate that is too high can lead to overfitting, and a learning rate that is too low can lead to underfitting. The learning rate should also be sensitive to the number of iterations in the algorithm. When increasing `num_iterations`, the learning rate should be decreased to control the contribution of each new tree. As indicated in Section 3.4, our dataset is of relatively small size, and the ratio between the number of features to the number of rows is quite high, which makes the model prone to overfitting. Therefore, we leaned toward testing learning rates at the lower end of the spectrum.

Hyperparameter	**Values**
Gradient Boosting Method	GOSS
L2 Regularization	(0, 0.1, 0.2, 0.3)
Early Stopping Rounds	(25, 50)
Number of Iterations	(16, 32, 64, 128, 256)
Number of Leaves	(8, 16, 32)
Maximum Depth	(-1, 8, 32)
Learning Rate	(0.01, 0.025, 0.05)

Table 10: The hyperparameter search space for LightGBM. All combinations of these hyperparameters were tested on the validation set prior to making the final decision on which hyperparameters to use.

3.5.3 Econometric Benchmarks

In order to compare the performance of AI models with econometric benchmarks, the AR, ARIMA, and ARIMAX models were employed.

An AR of order p, denoted as AR(p), can be expressed as the following equation:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \tag{1}$$

In an AR(p) model, the current value of the variable depends on its own values in the previous p periods, with each lagged value multiplied by its corresponding autoregressive coefficient. The term ε_t is a white noise disturbance term. Stationarity is an important quality for an estimated AR model because it helps prevent the model's past errors from having an increasingly significant impact on the current value of the variable over time (Brooks, 2019).

An ARMA(p,q) model combines the AR(p) and moving average MA(q) models, which means that the current value of a time series, y, is linearly dependent on both its own previous values and a combination of the current and past values of a white noise error term. This can be expressed as in the following equation:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2}$$

An ARIMA(p,d,q) model is simply an extension of the ARMA model where the d parameter represents the number of times the time series is differenced to achieve stationarity.

The AR and ARIMA models do not inherently provide explainability in regard to external factors, as they are designed to capture the internal dynamics of a time series. The ARIMAX model is an extension of the ARIMA model that incorporates exogenous variables.

In the ARIMAX(p,d,q) model with k exogenous variables, the exogenous factors will be captured through a linear relationship:

$$\sum_{j=1}^{k} \beta_j x_{j,t} \tag{3}$$

where $x_{j,t}$ represents the j-th exogenous variable at time t, and β_j is the coefficient of the j-th exogenous variable. This means that the coefficients β_j provide explainability as they represent the strength and direction of the exogenous variables' influence on the target variable. In general, the ARIMAX(p,d,q) model can be represented as:

$$d_t = \phi_0 + \sum_{i=1}^{p} \phi_k d_{t-i} + \varepsilon_t + \sum_{k=i}^{q} \theta_i \varepsilon_{t-i} + \sum_{j=1}^{k} \beta_j x_{j,t} \tag{4}$$

Where d_t is the d-th differenced series of the time series y_t, and x_{jt} is the j-th external explanatory variable, $j = 1, ..., k$.

Stationarity
The Augmented Dickey-Fuller (ADF) test, introduced by Dickey and Fuller (1981), is used to determine whether a time series has a unit root, indicating non-stationarity. The test estimates an autoregressive model, and the null hypothesis of the test is that the series has a unit root, i.e., it is non-stationary. The alternative hypothesis is that the series is stationary. If the series is non-stationary, differencing is performed and the test is iterated until stationarity is attained.

MA Lag Order
In the ARIMA and ARIMAX models, the Autocorrelation Function (ACF) plot can be used to identify the appropriate number of Moving Average (MA) terms. The ACF plot shows the correlation between a time series and its lagged values. A significant spike in the ACF plot at a particular lag indicates that there is a correlation between the series at time t and the same series at time t-k that is not accounted for by the intervening lags. This suggests that an MA term of order k may be appropriate for modeling the time series.

To determine the number of MA terms, the models were tested on the validation set with lags of order up to k. Then, the number of MA terms that gave the best performance on the validation set was selected. We chose not to try higher orders than k to avoid overfitting.

AR Lag Order
The Partial Autocorrelation Function (PACF) plot is a useful tool for identifying the number of AR terms in the AR, ARIMA, and ARIMAX models. The PACF plot displays the correlation between a time series and its lagged values after accounting for the correlation explained by the intervening lags. A significant spike in the PACF plot at lag k indicates that there is a correlation between the series at time t and the same series at time t-k, that is not accounted for by the intervening lags. This suggests that an AR term of order k may be appropriate for modeling the time series. Overall, the models were tested on the validation with lags up to order k.

Number of exogenous variables
To ensure a fair comparison, identical exogenous variables were used in the ARIMAX model as in the AI models. Specifically, these are the variables selected in Section 3.2, and shown in Table 5.

3.6 XAI: SHAP Framwork

SHAP is an XAI framework that was introduced by Lundberg and Lee (2017a), and builds on the concept of Shapley values of Shapley et al. (1953). SHAP is an additive feature attribution method, which means that it can be expressed as:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{5}$$

The explanation model g takes a simplified input feature vector $z' \in \{0,1\}^M$, where M is the number of features, and $\phi_i \in \mathbb{R}$ represents the attribution effect of feature i. To explain the contribution of individual features for a specific instance, the Shapley value can be used to calculate the average marginal contribution of each feature over all possible feature combinations. Lundberg and Lee (2017a) showed that there is a unique solution within the additive feature attribution methods that satisfies the desirable properties of local accuracy, missingness, and consistency. This solution can be expressed as follows:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f(x,z') - f(x, z' \setminus i)] \tag{6}$$

Here, f represents the original model we wish to explain and x denotes the original input. The term $|z'|$ is the number of non-zero elements in z'. $z' \in x'$ represents all vectors of z' where the non-zero elements are a subset of the non-zero elements in x'. In other words, the Shapley value can be used to construct a locally linear explanatory model that approximates the original model for a specific input x, representing local accuracy. Another key property of the Shapley value is that it is zero whenever a feature has no impact on the model, which accounts for missingness. Additionally, if a feature's contribution increases in a second model, its corresponding Shapley value will also increase, indicating consistency.

The SHAP values, as the solution to Equation 6, are interpreted as the Shapley values of a conditional expectation function of the original model. They indicate the change in expected prediction when conditioning on a particular feature. Waterfall plots are typically used to visualize local explanations, displaying feature contributions for individual predictions, while beeswarm plots can be used for global explanations in full model interpretation.

While the SHAP framework offers advantages over earlier XAI frameworks in terms of computational efficiency and consistency between local and global interpretations, it is based on the assumption of feature independence, which is often violated in real-world applications. To address this limitation, several alternative applications have been proposed to improve SHAP computation for specific models. The TreeSHAP implementation of Lundberg and Lee (2017a) assumes less dependence than similar methods. This implies that it considers certain dependencies, but not all of them (Aas et al., 2021). In this study, we use TreeSHAP to explain the results of LightGBM, and the general explainer interface with the algorithm set to `auto` for the neural network model.

3.7 Measuring Model Performance

To make meaningful comparisons between different models, it is essential to employ metrics to evaluate the performance of various techniques using a uniform approach. We used regression metrics for evaluating the prediction performance, along with classification metrics to evaluate the model's ability to predict the correct sign (upward or downward move). To evaluate the degree to which the models provide explainability, a qualitative discussion is provided in Section 4.3.

Using a single metric to evaluate the accuracy of a regression model is not sufficient, as it does not provide a complete picture of the model's performance. Employing multiple metrics is necessary to comprehensively understand the model's accuracy. We use Mean Absolute Error (MAE), Mean Squared Error (MSE) and bias to evaluate the regression fit. Additionally, we use accuracy to measure the models' ability to predict the correct sign.

MSE
The MSE is a metric used to measure the average squared difference between the predicted and actual values in a regression model. The MSE is calculated by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{7}$$

MSE tends to give more weight to large errors as it squares the difference between actual and predicted data.

MAE
The MAE measures the average absolute difference between the predicted and actual values in a regression model. The MAE is calculated by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8}$$

MAE treats all errors equally, regardless of whether they are large or small, and is less sensitive to outliers and large errors.

Bias
Bias refers to the systematic error that the model introduces in the prediction. It measures how far off, on average, the model's predictions are from the true values. A positive bias indicates that the model typically overpredicts, while a negative bias indicates that it typically underpredicts. A bias approaching zero implies that the errors primarily originate from variance, rather than from consistent, systematic errors (Hastie et al., 2017). The bias is calculated by:

$$\text{Bias} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) \tag{9}$$

Here y_i is the actual value for the i^{th} instance in the dataset, and $\hat{y}_i$ represents the predicted value for the i^{th} instance.

Classification

In addition to the regression performance measures, we apply a classification metric to evaluate the degree to which the model is able to classify whether the bond spread is going to increase or decrease (up or down moves). A correct prediction, in Equation (10) denoted as a true positive (TP) or true negative (TP) prediction, is when the model correctly predicts whether the next quarter will have a higher or lower spread compared to the current quarter. An incorrect prediction, a false positive (FP), or a false negative (FN), is when the direction of the prediction is wrong.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

The accuracy of a binary classification is simply the total number of correct predictions as a portion of the total number of predictions. As this is primarily a regression problem and as we have a measure for bias, we don't see it as necessary to use additional classification metrics beyond accuracy.

4 Results

This section begins with presenting the chosen hyperparameters for the models, determined based on the validation set. Next, the model's out-of-sample prediction performance on the test set is provided. Finally, the extent to which each model provides explainability is discussed.

4.1 Hyperparameter Tuning Results

The hyperparameter space, as defined for each model in Section 3.7, was tested on the validation set. For each model, the combination of hyperparameters that resulted in the lowest validation set MSE was chosen. In the event of an equal MSE between multiple runs, the least complex model was selected to mitigate the risk of overfitting.

4.1.1 Hyperparameter Tuning for ANN

Figure 15 in Appendix F shows hyperparameters and corresponding MSE as a result of 500 iterations of search throughout the hyperparameter space. Table 11 displays the selected hyperparameters for the ANN model for each country.

Hyperparameter	**Greece**	**Italy**	**Portugal**	**Spain**
Number of hidden layers	3	3	3	3
Number of units per hidden layer	32	64	256	128
Activation function (in hidden layers)	ReLU	ReLU	ReLU	ReLU
Learning rate	0.0156	0.0027	0.0052	0.0056
Dropout	0.6	0.8	0.7	0.8
Batch size	64	64	64	64
Epochs	64	128	256	256

Table 11: The final hyperparameters for ANN. The hyperparameters were chosen based on their ability to deliver the highest performance on the validation set.

The complexity of the neural network is generally determined by the number of hidden layers and the number of units within each layer. We note that 3 hidden layers were consistently favored across all countries. Regarding the number of units per layer, we observe that Greece preferred a relatively simple model, going for 32 units per layer. We argue that this preference can be attributed to the presence of extreme bond spreads in the training data for the country. Consequently, if the model had a more complex structure, it would have been vulnerable to overfitting.

Additionally, it is worth noting that a high dropout rate and a low learning rate were generally preferred across all countries. These choices contribute to reducing the model's vulnerability to overfitting, underscoring the importance of addressing overfitting concerns when working with a relatively small dataset.

4.1.2 Hyperparameter Tuning for LightGBM

Figure 16 in Appendix F shows hyperparameters and corresponding MSE as a result of 500 iterations of search throughout the hyperparameter space for the LightGBM model.

Table 12 displays the selected hyperparameters for the LightGBM model for each country.

Hyperparameter	Greece	Italy	Portugal	Spain
Gradient Boosting Method	GOSS	GOSS	GOSS	GOSS
L2 Regularization	0.0	0.3	0.3	0.2
Early Stopping Rounds	50	25	25	25
Number of Iterations	16	32	128	64
Number of Leaves	32	8	8	16
Maximum Depth	8	16	4	8
Learning Rate	0.05	0.05	0.05	0.05

Table 12: The final hyperparameters for LightGBM. The hyperparameters were chosen based on their ability to deliver the highest performance on the validation set.

The number of iterations serves as a clear indicator of the LightGBM model's complexity, with a new tree created at each iteration. We notice that simpler models performed well on the validation set, which was particularly evident in Greece and Italy. Specifically, a preference was given to 16 and 32 iterations in Greece and Italy, respectively. This observation aligns with the results from the ANN, where we also noticed a preference for simpler models in these countries.

We observe that a learning rate of 0.05 was preferred consistently across all countries. This learning rate, being the highest one tested, indicates the importance of the model learning from recent experiences when forecasting the bond spread for the subsequent quarter. The remaining hyperparameters varied across the countries. We believe this could be linked to the fact that different parameters work well with the selected number of iterations depending on the country-specific context.

4.1.3 Hyperparameter Tuning for Econometric Benchmarks

Differencing

In Table 13, the p-values obtained from the ADF test for the four countries are presented. For the original time series data, all p-values exceed 0.05, which is a commonly used significance level, indicating that the null hypothesis of non-stationarity can not be rejected. However, after differencing the time series data, all p-values are below $1 * 10^{-7}$, providing strong evidence that the data is stationary.

Country	Original data	1st difference
Greece	0.195	$1.213 * 10^{-19}$
Italy	0.258	$1.645 * 10^{-17}$
Portugal	0.264	$5.359 * 10^{-8}$
Spain	0.556	$2.737 * 10^{-14}$

Table 13: P-values from ADF test on the target variable in each country. The findings indicate that the null hypothesis of non-stationarity cannot be rejected for the original data, but it can be rejected when considering the first difference in the data.

Hence, the time series data were subjected to a first-order differencing operation prior to constructing the AR model. Similarly, the differencing parameter was set to 1 for the ARIMA and ARIMAX models.

AR Order
The PACF plots for the four countries are displayed in Figure 14 in Appendix E. The PACF plots reveal that Italy, Greece, and Spain have two notable spikes, while Portugal has three significant spikes. As a result, AR orders of up to 3 were evaluated on the validation sets.

MA Order
The ACF plots for the four countries are displayed in Figure 13 in Appendix E. The ACF plots indicate that Greece and Portugal have five significant spikes, while Italy and Spain have six notable spikes. Accordingly, MA orders up to 6 were evaluated on the validation set.

Final Parameters
The chosen orders for each model are presented in Table 14. The orders were selected based on the MSE for the validation set. Despite identifying 5 and 6 significant spikes on the ACF plots, the best performance on the validation set was achieved with an MA order of 1. Notably, low-order models generally showed better performance on the validation data, potentially because higher-order models are prone to overfitting.

Model	AR Order	Differencing	MA Order
AR Greece	2	1	NA
AR Italy	3	1	NA
AR Portugal	1	1	NA
AR Spain	2	1	NA
ARIMA Greece	2	1	1
ARIMA Italy	2	1	1
ARIMA Portugal	1	1	1
ARIMA Spain	2	1	1
ARIMAX Greece	2	1	1
ARIMAX Italy	2	1	1
ARIMAX Portugal	2	1	1
ARIMAX Spain	2	1	1

Table 14: Final hyperparameters for the econometric models. The hyperparameters that had the lowest MSE on the validation set were selected. Despite the indication of a higher MA order in the ACF plot, the models with an MA order of 1 demonstrated the best performance on the validation set.

4.2 Prediction Results

All prediction plots for the models can be found in Appendix G.

MSE
tBased on the results presented in Table 15, the AR model showed the highest average performance in regards to MSE, followed by LightGBM, ARIMA, ARIMAX, and ANN.

With the exception of ANN, all models demonstrated similar performance levels. On average, ANN performed worse compared to the other models. The predictions generated by ANN can be observed in Figure 17 in Appendix G, revealing the challenges it faced in finding the right balance between underfitting and overfitting the data. Specifically, the ANN model appeared to underfit the data for Spain while overfitting it for Greece. These issues are commonly encountered by neural networks when dealing with relatively small

Model	Greece	Italy	Portugal	Spain	Average
ANN	2.245	0.441	0.700	0.049	0.859
LightGBM	0.426	0.216	0.342	0.055	0.260
AR	0.575	0.221	0.164	0.059	**0.255**
ARIMA	0.606	0.221	0.165	0.063	0.264
ARIMAX	0.762	0.199	0.196	0.055	0.303

Table 15: MSE results by model and country. On average, the AR model demonstrated the highest performance, followed by LightGBM, ARIMA, ARIMAX, and ANN.

datasets. Conversely, the LightGBM model consistently showed comparable performance to the benchmark models in terms of prediction performance. It is interesting that the AR model showed the best performance, raising doubts about whether external variables enhance predictive capabilities.

MAE

According to the MAE results shown in Table 16, the AR model exhibited the highest average performance, followed by ARIMA, LightGBM, ARIMAX, and ANN.

Model	Greece	Italy	Portugal	Spain	Average
ANN	1.006	0.541	0.581	0.192	0.572
LightGBM	0.574	0.362	0.368	0.186	0.372
AR	0.552	0.325	0.288	0.177	**0.336**
ARIMA	0.572	0.318	0.279	0.178	0.337
ARIMAX	0.701	0.323	0.330	0.168	0.381

Table 16: MAE results by model and country. On average, the AR model demonstrated the highest performance, followed by ARIMA, LightGBM, ARIMAX, and ANN.

The results with respect to MAE align with those obtained using MSE. Regarding the ranking of models based on average performance, the only distinction is that the ARIMA model outperforms the LightGBM model in this context. Since MSE is more responsive to large errors than MAE, it implies that the ARIMA model occasionally shows more severe errors for individual data points. Nevertheless, given the limited number of observations and the marginal difference in performance between the ARIMA and LightGBM models, it is not justifiable to make a definitive conclusion regarding their comparison based on these findings alone. Although the performance of the ANN is still noticeably worse than the other models, it is worth highlighting that the gap decreases when evaluating based on MAE. This is largely due to the fact that the substantial errors the ANN model produced for Greece are not penalized to the same extent by MAE.

Bias

The Bias results are shown in Table 17. The LightGBM model has the lowest absolute average bias across all the countries, while the ANN model has the highest absolute average.

The ANN model exhibits a significant overestimation of values for Greece while displaying a negative bias for all other countries. The other models generally demonstrate a positive bias, indicating overestimation in their predictions. By comparing the value of the bias with the value of the MAE, it becomes evident that the bias constitutes a relatively small portion of the overall error for the LightGBM model. However, for all the other models, the bias represents a substantial portion of the overall error.

Model	Greece	Italy	Portugal	Spain	Average	Absolute Average
ANN	0.996	-0.360	-0.577	-0.174	-0.029	0.527
LightGBM	-0.049	0.039	0.097	0.109	0.049	**0.074**
AR	0.474	-0.001	0.156	0.042	0.168	0.168
ARIMA	0.506	-0.001	0.150	0.035	0.173	0.173
ARIMAX	0.280	0.045	0.243	0.047	0.154	0.154

Table 17: Bias results by model and country. In terms of average absolute bias, the LightGBM model demonstrates the lowest bias, followed by ARIMAX, AR, ARIMA, and ANN in descending order.

Directional Classification Accuracy

In terms of classification accuracy, there was a notable shift in the average rankings of the models. The ANN model exhibited the highest performance, with LightGBM, ARIMA, ARIMAX, and AR following in descending order as shown in Table 18.

Model	Greece	Italy	Portugal	Spain	Average
ANN	0.833	0.583	0.667	0.583	**0.667**
LightGBM	0.583	0.667	0.667	0.500	0.604
AR	0.583	0.583	0.583	0.500	0.562
ARIMA	0.583	0.667	0.583	0.500	0.583
ARIMAX	0.500	0.667	0.583	0.583	0.583

Table 18: Directional classification accuracy results by model and country. On average, the ANN model demonstrated the highest performance, followed by LightGBM, ARIMA, ARIMAX, and AR.

It is noteworthy that the top two models in this context are both AI models. This observation suggests that they are proficient in capturing the directional movement, although they may occasionally miss out on accurately predicting the magnitude of the movement.

Across the three metrics for prediction performance, the AI models generated noteworthy results. The LightGBM model demonstrated comparable performance to the benchmark models in terms of both MAE and MSE. Additionally, it slightly outperformed the benchmark models in directional classification accuracy. On the other hand, the ANN model exhibited worse performance compared to the benchmark models in terms of both MSE and MAE. However, it showed better performance in classification accuracy.

As noted in Section 2.3, de Oliveira and Montes (2021) and Bianchi et al. (2021) both compared tree-based models and perceptron-based ML models in a similar context. Bianchi et al. (2021) found that both neural networks and extreme trees are appropriate to predict the return of sovereign bonds. However, our results differ from the study by de Oliveira and Montes (2021), where their Multilayer Perceptron (MLP) model performed better than their GBDT model. We note that our study used different model implementations and focused on a different market compared to de Oliveira and Montes (2021).

4.3 Explainability Results

For the AI models, explainability results are provided as SHAP beeswarm and waterfall plots representing global and local explainability, respectively. In addition, the explainability of the benchmark ARIMAX model is provided by the parameter coefficients of the trained model.

4.3.1 ANN Explainability Results

Figure 3 presents the SHAP values of the 10 most impactful features for the ANN model, separated by country. The top variables across all countries include the two preceding lags of the bond spread, GDP per capita, and monetary freedom. Except for monetary freedom, these variables are all among the most used variables in the literature (see Section 2.2). The figure illustrates that higher past values of bond spread contribute to a high target bond spread. Simultaneously, a lower degree of monetary freedom and GDP per capita is associated with an increased spread. The importance of other variables tends to vary from country to country. However, unemployment, government spending, and the CPI are commonly observed as important factors.

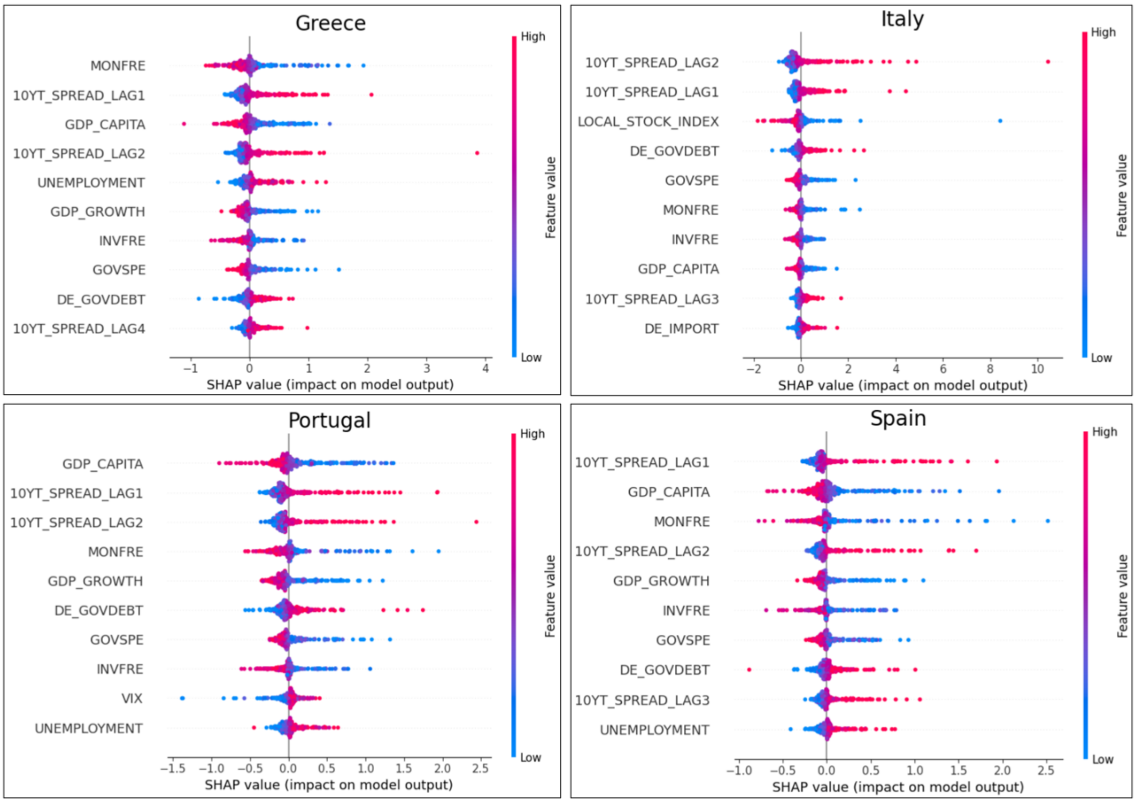


Figure 3: Top 10 global features based on SHAP values from the ANN model by country. The two preceding lags of bond spread, monetary freedom, and GDP per capita are identified as consistently being among the most important features across all countries. Explanations for the variable names are provided in Table 5.

Another observation from Figure 3 is the presence of somewhat stretched lines, complicating definitive conclusions about the exact influence or magnitude of a specific variable. We argue that this complexity is likely to arise due to the highly interconnected structure of the neural network, making its interpretation more challenging. Nevertheless, it remains feasible to draw conclusions regarding the direction or sign of the influence for each variable.

Local explainability for individual predictions is provided by the model through the use of waterfall plots, exemplified for the last prediction (Q4 2019) in Figure 4. The local explainability is consistent with the global explainability for certain variables, such as GDP per capita. However, the local explainability does not emphasize the significance of the previous lags of bond spreads to the same extent, despite the established importance of these variables. This may indicate that the ANN model is not that reliable on the local prediction level, as it may have overfitted on the data.

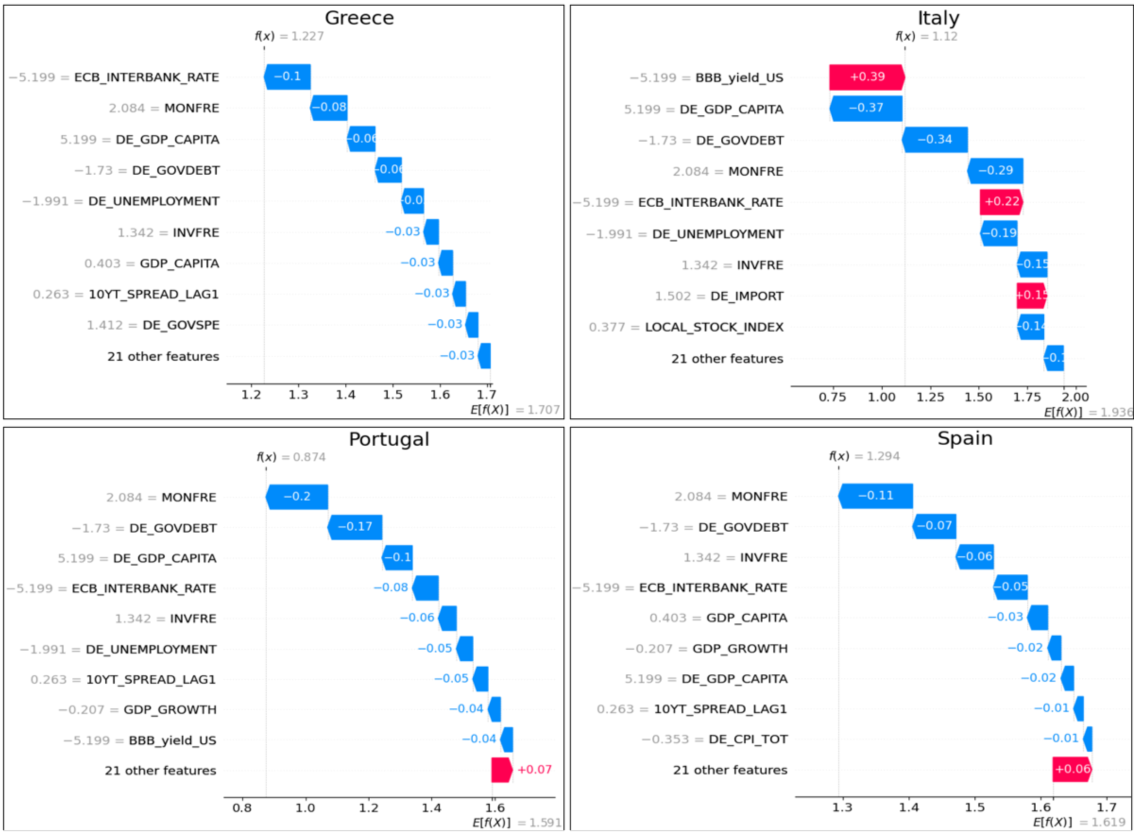


Figure 4: SHAP local explainability for the ANN model. The waterfall plot provides the local explainability for the last prediction (Q4 2019) in each country, offering explainability specifically for those particular predictions.

4.3.2 LightGBM Explainability Results

Figure 5 displays the SHAP values for the ten most important features globally in the LightGBM model, organized by country.

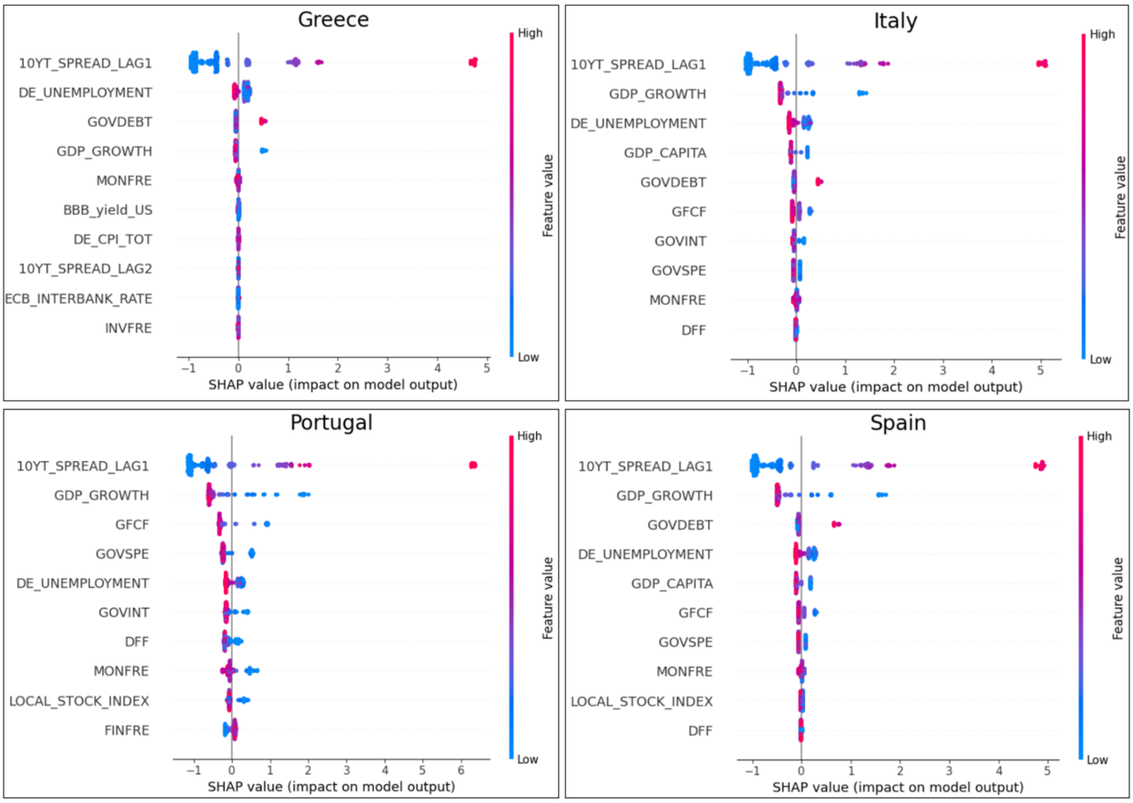


Figure 5: Top 10 global features based on SHAP values from the LightGBM model by country. Across all countries, the first lag of the bond spread and GDP growth consistently show up as important features. Explanations for the variable names are provided in Table 5.

In regards to global explainability, we note that even though two lags of the bond spread were of high importance in the ANN model, it is primarily the first lag that emerges as important in this context. Furthermore, the LightGBM model underscores the importance of GDP per capita, but it indicates that GDP growth may be of even greater importance. The model associates low GDP growth with a greater spread. Government debt also emerges as a commonly important factor, with high debt levels correlating with a wider spread.

As seen in Figure 5, the LightGBM model yields fewer variables of high importance compared to the ANN. Furthermore, the compressed lines corresponding to each variable simplify the task of understanding their exact impact.

Local explainability for individual predictions is provided through the use of waterfall plots, as exemplified in Figure 6 for the Q4 2019 prediction. Overall, the local explainability plot provides clear and easily understandable reasoning for the LightGBM models' predictions. The results of the local explainability align with the global explainability, highlighting many of the same features. Furthermore, we note that the number of features that appear to have an impact on the prediction varies across countries. For example, only two variables (`10_yt_spread_lag1` and `de_unemployment`) seem to explain most of the prediction for Greece. This can be attributed to the fact that the model for Greece has the lowest complexity based on the selected hyperparameters, as discussed in Section 4.1.2.

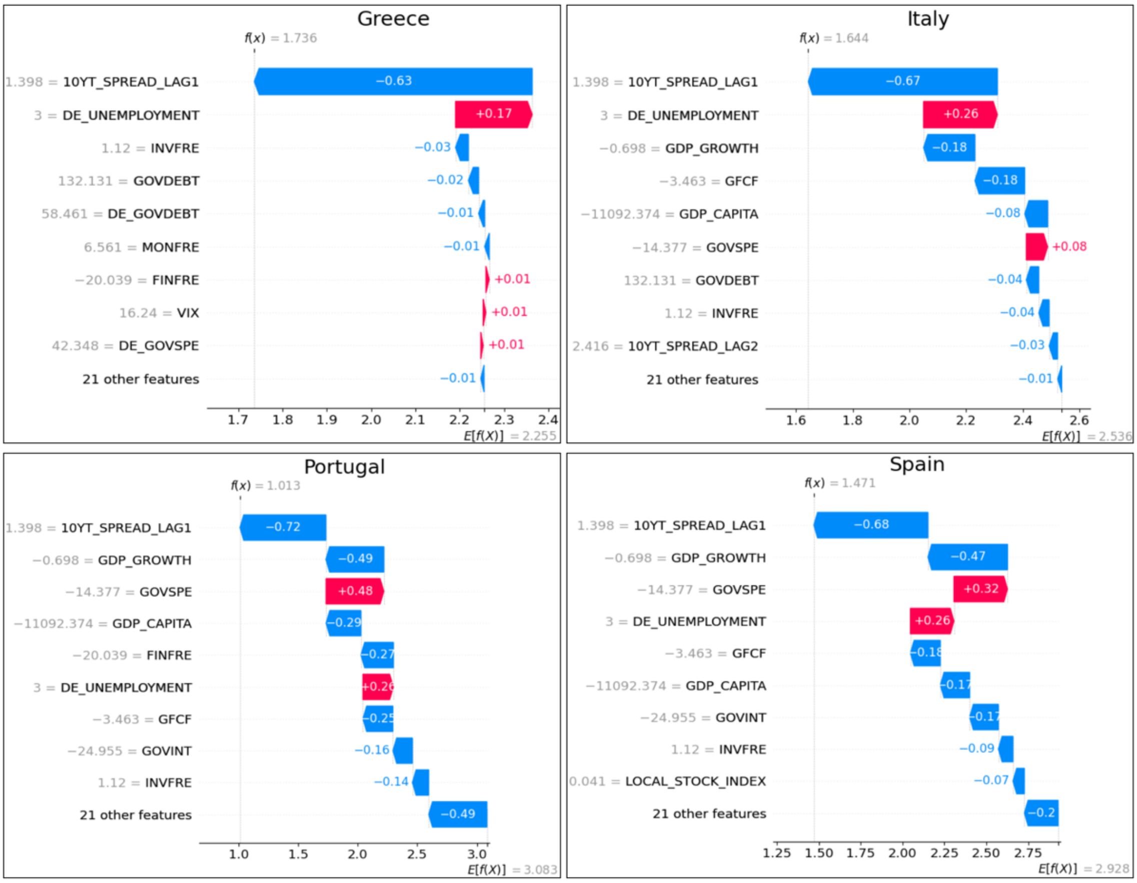
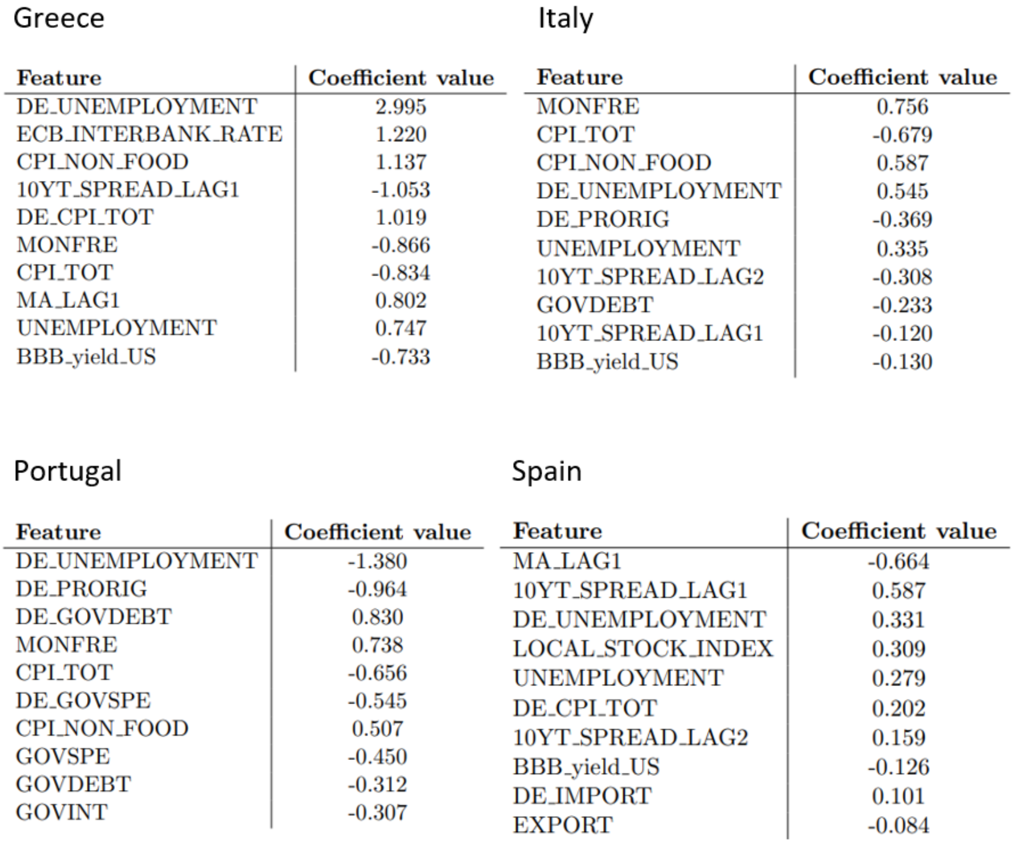


Figure 6: SHAP local explainability for the LightGBM model. The waterfall plot provides the local explainability for the last prediction (Q4 2019) in each country, offering explainability specifically for those particular predictions.

4.3.3 ARIMAX Explainability Results

Among the benchmark models, the ARIMAX model is the only one to offer explainability in regard to exogenous variables. Figure 7 shows the ten most significant features, based on the absolute value of their coefficients in the fitted model, categorized by country. Compared to the AI models, the importance of variables varies significantly from one country to another. However, the lag of the bond spread, unemployment, and CPI consistently appear as important variables. The importance of the variables has substantial variation, not only in the magnitude but also in the sign of their values. This can be attributed to the covariation among the variables and the inclusion of many features. For instance, in the case of Italy, we observe that the total CPI has a notable negative impact on the prediction, whereas the CPI for non-food and -energy has a positive influence. To mitigate this concern, employing a more rigorous feature selection process would have been beneficial. This could have resulted in the exclusion of certain highly correlated variables. However, it is worth noting that the SHAP values from the AI models were able to provide insightful explanations using the same set of features without the need to modify the correlated variables or conduct additional tests.



Greece

Feature	Coefficient value
DE_UNEMPLOYMENT	2.995
ECB_INTERBANK_RATE	1.220
CPI_NON_FOOD	1.137
10YT_SPREAD_LAG1	-1.053
DE_CPI_TOT	1.019
MONFRE	-0.866
CPI_TOT	-0.834
MA_LAG1	0.802
UNEMPLOYMENT	0.747
BBB_yield_US	-0.733

Italy

Feature	Coefficient value
MONFRE	0.756
CPI_TOT	-0.679
CPI_NON_FOOD	0.587
DE_UNEMPLOYMENT	0.545
DE_PRORIG	-0.369
UNEMPLOYMENT	0.335
10YT_SPREAD_LAG2	-0.308
GOVDEBT	-0.233
10YT_SPREAD_LAG1	-0.120
BBB_yield_US	-0.130

Portugal

Feature	Coefficient value
DE_UNEMPLOYMENT	-1.380
DE_PRORIG	-0.964
DE_GOVDEBT	0.830
MONFRE	0.738
CPI_TOT	-0.656
DE_GOVSPE	-0.545
CPI_NON_FOOD	0.507
GOVSPE	-0.450
GOVDEBT	-0.312
GOVINT	-0.307

Spain

Feature	Coefficient value
MA_LAG1	-0.664
10YT_SPREAD_LAG1	0.587
DE_UNEMPLOYMENT	0.331
LOCAL_STOCK_INDEX	0.309
UNEMPLOYMENT	0.279
DE_CPI_TOT	0.202
10YT_SPREAD_LAG2	0.159
BBB_yield_US	-0.126
DE_IMPORT	0.101
EXPORT	-0.084

Figure 7: Top 10 feature coefficient values from the ARIMAX model by country. The moving average terms are represented by the MA_LAG features, whereas the autoregressive terms are denoted as 10YT_SPREAD_LAG. Explanations for the exogenous variables are provided in Table 5.

5 Discussion and Conclusion

This study examined the prediction of sovereign bond spreads by assessing AI models and econometric benchmarks. Specifically, this research employed the ANN and LightGBM models as AI approaches, while the AR, ARIMA, and ARIMAX models served as econometric benchmarks. The evaluation covered both prediction accuracy and explainability, leveraging the use of SHAP, an XAI technique.

In terms of predictive performance, the LightGBM model demonstrated comparable results to the benchmark models, while the ANN model demonstrated lower performance. Figure 17 illustrates that the ANN model encountered challenges in striking the appropriate balance between overfitting and underfitting. It is known that ANN models typically dominate with larger datasets, and our findings support the hypothesis that quarterly data within this restricted timeframe was not optimal for the ANN model. The exception to this trend was observed in terms of directional classification accuracy, where the ANN model outperformed all other models. Hence, it appears that the ANN model demonstrates a good understanding of directional movement but faces challenges when it comes to capturing the magnitude. The LightGBM model also outperformed the benchmarks in terms of classification accuracy, showcasing a particularly strong use case for the ML models.

The LightGBM model demonstrated the capacity to offer meaningful global and local explanations for its predictions using the SHAP framework. To some extent, this holds true for the ANN model as well. However, the ANN model suggested a higher number of important variables, making the exact impact of each variable less clear. In addition, the ANN model did not highlight the importance of the previous lags of the target variable as clearly as the LightGBM model did, in terms of local explainability. The importance of these variables is widely recognized, so this constitutes an issue for the ANN model.

The limitations of the ANN model, in terms of both prediction accuracy and explainability, can be attributed to its complex structure. ANN models are known to perform better with larger datasets but face challenges in this environment with quarterly data. In contrast, the LightGBM model proves to be well-suited, providing both explainability and accurate predictions with its simpler tree-based structure. As stated in Section 2.4, Aas et al. (2021) argue that Shapley values can produce misleading explanations when features are correlated. Another concern with SHAP values is that while they offer a transparent indication of variable importance, they do not clarify the true statistical significance of these variables. This represents an area for additional research and poses a potential limitation in terms of the reliability of our explainability results.

The exogenous variables identified as important in both of the AI models and across all countries were GDP per capita, GDP growth, CPI, government debt, and unemployment. Notably, the first four variables were found to be among the most frequently used in the literature, as discussed in our literature review in Section 2.2. Although unemployment was not among the variables we found to be the most used, it remains commonly employed, making these results unsurprising. GDP growth is a variable that has been subject to debate regarding its significance in the literature as discussed in Section 2.2. For instance, Maltritz (2012) and Maltritz and Molchanov (2013) found the variable to be insignificant. However, our LightGBM model highlights the high importance of GDP growth, while the importance of this variable in the ANN model's results is less conclusive. This represents one of several instances where the models diverge in terms of the importance assigned to variables, highlighting a limitation in our findings. Nonetheless, we maintain the belief that individual results should not necessarily undermine others, as the effectiveness of variables can vary depending on the specific model in use.

To put our results regarding the models' predictive power into a wider context, we can first consider the concept known as the spanning hypothesis. The spanning hypothesis argues that information held in the yield curve itself is sufficient for forecasting bond yields. As noted by Huang and Shi (2023), there is no consensus in the literature as to whether macro variables have incremental predictive power for bond yields. Nonetheless, their results suggest that a linear combination of a given set of macro factors is able to provide additional predictive power, contradicting the spanning hypothesis. Our results demonstrate that the models with no exogenous variables, the AR and ARIMA models, perform the best on the MSE and MAE measures. This provides support for the spanning hypothesis. However, as already discussed, the ML models outperform the AR and ARIMA models in directional classification. The ML models also emphasize the importance of some exogenous variables, providing evidence contradicting the spanning hypothesis.

Regarding the performance of ML compared to statistical methods in time series forecasting, Makridakis et al. (2018) found ML methods to perform worse than simple statistical methods. Our findings demonstrate similar indications, as the ML methods are outperformed by both AR and ARIMA models on multiple measures. However, Makridakis et al. (2023) highlight recent advancements in ML methods for forecasting time series, such as hybrid methods and ML ensemble methods. Hybrid methods combine ML and traditional statistical methods and have recently been outperforming both ML and statistical methods alone. We find hybrid methods to be an interesting research area, as our results indicate that different models have different areas of strength. However, more complex models may come at the expense of explainability, training times, and resource use.

6 Limitations and Further Research

We have identified several limitations and areas that offer potential for future improvement in our research. Specifically, six areas of interest have emerged: (i) the inclusion of higher frequency data, (ii) conducting a more thorough analysis of explainability in benchmark models, (iii) exploring time-dependent effects such as structural breaks, (iv) expanding the variables by including unconventional factors such as news, (v) performing sensitivity analyses and estimating confidence intervals for the predictions, and (vi) the use of complex ML ensemble methods and hybrid methods.

First, to better capture the dynamics of sovereign bond spreads, a more comprehensive dataset with a greater number of data points would be advantageous. The lack of such data appears to have had a notable impact, particularly on the performance of the ANN model. Second, we encountered difficulties in extracting interpretability from the ARIMAX benchmark model to the same extent as the AI models. However, it does not mean we consider it impossible. We recognize that implementing a more rigorous feature selection process could have improved the explainability. Tests can also be performed to determine the statistical significance of the variables. However, it is important to note that the benchmark models were not the primary focus of this study.

Third, we have not explored the influence of structural breaks on bond spread prediction models over time. Exploring this aspect is crucial, especially in post-2020 predictions, due to the impact of asset purchase programs during the Covid-19 pandemic. These policies may have caused notable structural shifts in the bond markets. Fourth, as stated in Section 2.1, some recent studies suggest using media and news to predict sovereign bond spreads. For instance, Milas et al. (2021) found that tweets can improve predictions. Hence, the inclusion of media effects and other additional variables presents a potential improvement to our model.

Fifth, our study concentrates exclusively on the predicted values, without conducting sensitivity analyses or establishing confidence intervals. We believe that Bayesian Neural Network (BNN), as introduced by MacKay (1992), represents a feasible method to accomplish this objective. Sixth, hybrid methods and ML ensemble methods have shown good potential in forecasting time series data. Thus, the application of these methods on sovereign bond spreads constitutes an interesting area for further research.

Addressing these areas for improvement can lead to a better understanding of ML and XAI in the field of sovereign bond spread predictions.

Bibliography

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

Lars L Ankile and Kjartan Krange. Eikon python api wrapper. https://github.com/Krankile/eikon-wrapper/, 2022.

Massimo Aria and Corrado Cuccurullo. bibliometrix: An r-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4):959–975, 2017. URL https://doi.org/10.1016/j.joi.2017.08.007.

Vivek Arora and Martin Cerisola. How does us monetary policy influence sovereign spreads in emerging markets? *IMF Staff papers*, 48(3):474–498, 2001.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

Alena Audzeyeva and Ana-Maria Fuertes. On the predictability of emerging market sovereign credit spreads. *Journal of International Money and Finance*, 88:140–157, 2018.

Alessandro Beber, Michael W Brandt, and Kenneth A Kavajecz. Flight-to-quality or flight-to-liquidity? evidence from the euro-area bond market. *The Review of Financial Studies*, 22(3):925–957, 2009.

Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

Daniele Bianchi, Matthias Büchner, and Andrea Tamoni. Bond risk premiums with machine learning. *The Review of Financial Studies*, 34(2):1046–1089, 2021.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Steven A Block and Paul M Vaaler. The price of democracy: sovereign risk ratings, bond spreads and political business cycles in developing countries. *Journal of International Money and Finance*, 23(6):917–946, 2004.

Chris Brooks. *Introductory econometrics for finance*. Cambridge university press, 2019.

Sarah M Brooks, Raphael Cunha, and Layna Mosley. Categories, creditworthiness, and contagion: How investors' shortcuts affect sovereign debt markets. *International studies quarterly*, 59(3):587–601, 2015.

Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57:203–216, 2021.

François Chollet. Keras. https://github.com/keras-team/keras, 2015.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Pierre Collin-Dufresne and Robert S Goldstein. Do credit spreads reflect stationary leverage ratios? *The journal of finance*, 56(5):1929–1957, 2001.

Fabio Comelli. Emerging market sovereign bond spreads: Estimation and back-testing. *Emerging Markets Review*, 13(4):598–625, 2012.

Sergio Consoli, Luca Tiozzo Pezzoli, and Elisa Tosetti. Emotions in macroeconomic news and their impact on the european bond market. *Journal of International Money and Finance*, 118:102472, 2021.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Filipa Da Silva Fernandes, Charalampos Stasinakis, and Zivile Zekaite. Forecasting government bond spreads with heuristic models: evidence from the eurozone periphery. *Annals of Operations Research*, 282(1):87–118, 2019.

Leo De Haan, Jeroen Hessel, and Jan Willem van den End. Are european sovereign bonds fairly priced? the role of modelling uncertainty. *Journal of International Money and Finance*, 47:239–267, 2014.

Diego Silveira Pacheco de Oliveira and Gabriel Caldas Montes. Forecasting sovereign risk perception of brazilian bonds: an evaluation of machine learning prediction accuracy. *International Journal of Emerging Markets*, 2021.

Lara Marie Demajo, Vince Vella, and Alexiei Dingli. Explainable ai for interpretable credit scoring. *arXiv preprint arXiv:2012.03749*, 2020.

David A Dickey and Wayne A Fuller. Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica: journal of the Econometric Society*, pages 1057–1072, 1981.

Francis X Diebold and Canlin Li. Forecasting the term structure of government bond yields. *Journal of econometrics*, 130(2):337–364, 2006.

Francis X Diebold and Jose A Lopez. *Modeling volatility dynamics*. Springer, 1995.

Darrell Duffie. Credit risk modeling with affine processes. *Journal of Banking & Finance*, 29(11):2751–2802, 2005.

Christina Erlwein-Sayer. Macroeconomic news sentiment: enhanced risk assessment for sovereign bonds. *Risks*, 6(4):141, 2018.

ECB European Central Bank. *Financial Stability Review*. European Central Bank, Frankfurt am Main, Germany, November 2022.

ECB European Central Bank. Ecb stages of economic and monetary union (emu). https://www.ecb.europa.eu/ecb/history/emu/html/index.en.html, 2023. Accessed: 2023-05-24.

Eugene F Fama and Robert R Bliss. The information in long-maturity forward rates. *The American Economic Review*, pages 680–692, 1987.

Rui Fan, Oleksandr Talavera, and Vu Tran. Social media bots and stock markets. *European Financial Management*, 26(3):753–777, 2020.

FRED Federal Reserve Bank of St. Louis. Federal reserve economic data, 2023. URL https://fred.stlouisfed.org/.

FRED FRED Python Development Team. Fred python api. https://pypi.org/project/fredapi/, 2021. Accessed: 2023-03-01.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT'95 Barcelona, Spain, March 13–15, 1995 Proceedings 2*, pages 23–37. Springer, 1995.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R.* Spinger, 2013.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2017.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Jing-Zhi Huang and Zhan Shi. Machine-learning-based return predictors and the spanning controversy in macro-finance. *Management Science*, 69(3):1780–1804, 2023.

IMF International Monetary Fund. World Economic Outlook. IMF, Washington, D.C., 2023. URL https://www.imf.org/en/Publications/WEO. Accessed on May 29, 2023.

Gergana Jostova. Predictability in emerging sovereign debt markets. *The Journal of Business*, 79(2):527–565, 2006.

Uday Kamath and John Liu. *Explainable artificial intelligence: An introduction to interpretable machine learning.* Springer, 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

M Kearns. Thoughts on hypothesis boosting, ml class project. 1988.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Wanan Liu, Hong Fan, and Meng Xia. Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Systems with Applications*, 189:116034, 2022.

Francis A Longstaff, Sanjay Mithal, and Eric Neis. Corporate yield spreads: Default risk or liquidity? new evidence from the credit default swap market. *The journal of finance*, 60(5):2213–2253, 2005.

Francis A Longstaff, Jun Pan, Lasse H Pedersen, and Kenneth J Singleton. How sovereign is sovereign credit risk? *American Economic Journal: Macroeconomics*, 3(2):75–103, 2011.

Scott M Lundberg and Su-In Lee. Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060*, 2017a.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3): e0194889, 2018.

Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos, Artemios-Anargyros Semenoglou, Gary Mulder, and Konstantinos Nikolopoulos. Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward. *Journal of the Operational Research Society*, 74(3):840–859, 2023.

Dominik Maltritz. Determinants of sovereign yield spreads in the eurozone: A bayesian approach. *Journal of International Money and Finance*, 31(3):657–672, 2012.

Dominik Maltritz and Alexander Molchanov. Analyzing determinants of bond yield spreads with bayesian model averaging. *Journal of Banking & Finance*, 37(12):5275–5284, 2013.

Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

Microsoft. Lightgbm, 2023. URL https://lightgbm.readthedocs.io/en/latest/index.html.

Costas Milas, Theodore Panagiotidis, and Theologos Dergiades. Does it matter where you search? twitter versus traditional news media. *Journal of Money, Credit and Banking*, 53(7):1757–1795, 2021.

Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pages 1–66, 2022.

Karim El Mokhtari, Ben Peachey Higdon, and Ayşe Başar. Interpreting financial time series with shap values. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pages 166–172, 2019.

Saji Thazhugal Govindan Nair. Sovereign credit ratings and bond yield spreads in emerging markets: Revisiting cantor–packer evidence in resilience. *Journal of Financial Economic Policy*, 12(2):263–277, 2019.

Jun Pan and Kenneth J Singleton. Default and recovery implicit in the term structure of sovereign cds spreads. *The Journal of Finance*, 63(5):2345–2384, 2008.

Rahul Parhi and Robert D Nowak. The role of neural network activation functions. *IEEE Signal Processing Letters*, 27:1779–1783, 2020.

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

Hongyi Qian, Baohui Wang, Minghe Yuan, Songfeng Gao, and You Song. Financial distress prediction using a corrected feature selection measure and gradient boosted decision tree. *Expert Systems with Applications*, 190:116202, 2022.

Butch Quinto. *Next-Generation Machine Learning with Spark*. Springer, 2020. doi: 10.1007/978-1-4842-5669-5. URL https://link.springer.com/book/10.1007/978-1-4842-5669-5.

Refinitiv. Refinitiv eikon, 2023. URL https://www.refinitiv.com/en/products/eikon-trading-software.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Pedro Pires Ribeiro, Rodolfo Cermeño, and José Dias Curto. Sovereign bond markets and financial volatility dynamics: Panel-garch evidence for six euro area countries. *Finance Research Letters*, 21:107–114, 2017.

Matthew Richardson and Pedro Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. *Advances in neural information processing systems*, 14, 2001.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Isaac Jacob Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part b. on the problem of osculatory interpolation. a second class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(2):112–141, 1946.

SciPy SciPy community. *Scipy Interpolate CubicSpline*. SciPy community, 2021. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html.

Lloyd S Shapley et al. A value for n-person games. 1953.

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Xiaolei Sun, Mingxi Liu, and Zeqian Sima. A novel cryptocurrency price trend forecasting model based on lightgbm. *Finance Research Letters*, 32:101084, 2020.

William R Swartout and Johanna D Moore. Explanation in second generation expert systems. In *Second generation expert systems*, pages 543–585. Springer, 1993.

Heritage The Heritage Foundation. Index of economic freedom, 2023. URL https://www.heritage.org/index/.

Ellen Tobback, Hans Naudts, Walter Daelemans, Enric Junqué de Fortuny, and David Martens. Belgian economic policy uncertainty index: Improvement through text mining. *International journal of forecasting*, 34(2):355–365, 2018.

Paul M Vaaler, Burkhard N Schrage, and Steven A Block. Counting the investor vote: Political business cycle effects on sovereign bond spreads in developing countries. *Journal of international business studies*, 36(1):62–88, 2005.

Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Aabhas Vij and Preethi Nanjundan. Comparing strategies for post-hoc explanations in machine learning models. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2021*, pages 585–592. Springer, 2022.

Rashmi Korlakai Vinayak and Ran Gilad-Bachrach. Dart: Dropouts meet multiple additive regression trees. In *Artificial Intelligence and Statistics*, pages 489–497. PMLR, 2015.

WoS Web of Science. Web of science search. https://www.webofscience.com/wos/woscc/summary/6b78ec4a-29f3-4aa7-9880-4822dcc6c48d-5a77c40b/relevance/1, 2022. Accessed: 2022-10-31.

Michael R Wick and William B Thompson. Reconstructive expert system explanation. *Artificial Intelligence*, 54(1-2):33–70, 1992.

WB World Bank. *Global Economic Prospects*. World Bank, Washington, DC, 2023.

Shuxiang Xu and Ling Chen. A novel approach for determining the optimal number of hidden layer neurons for fnn's and its application in data mining. 2008.

Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

Dongyang Zhang and Yicheng Gong. The comparison of lightgbm and xgboost coupling factor analysis and prediagnosis of acute liver failure. *IEEE Access*, 8:220990–221003, 2020.

Appendix

A Selecting Papers for Literature Review on Sovereign Bond Spread Prediction

To identify relevant papers for the literature review on sovereign bond spread prediction, a combination of bibliometric methods and manual review was employed. To initiate the paper identification process, a search for *sovereign bond* spread** on Web of Science (WoS) yielded 614 results (Web of Science, 2022). The search for *sovereign bond* spread* predict** identified a subset of 61 papers specifically focusing on the prediction of sovereign bond spreads.

To gain a comprehensive understanding of the field and support paper selection, a co-citation analysis was applied. Bibliometrix, an R library, was employed for co-citation analysis, and the co-citation network was clustered using the Louvain algorithm (Aria and Cuccurullo, 2017). The PageRank algorithm by Richardson and Domingos (2001) was used to rank the papers based on citation influence. Table 19 displays the results of the PageRank algorithm, while Figure 8 illustrates the co-citation network.

Paper	Pagerank Score
Longstaff et al. (2011)	0.00401
Pan and Singleton (2008)	0.00282
Longstaff et al. (2005)	0.00273
Duffie (2005)	0.00255
Beber et al. (2009)	0.00242
Collin-Dufresne and Goldstein (2001)	0.00217
Block and Vaaler (2004)	0.00181
Diebold and Li (2006)	0.00176
Diebold and Lopez (1995)	0.00176
Fama and Bliss (1987)	0.00173

Table 19: Top 10 papers on their PageRank score. The PageRank score measures the centrality of the paper based on the co-citation network.

Subsequently, we conducted a manual review of the papers identified by the PageRank algorithm, as well as the 61 papers specifically focusing on prediction obtained from WoS. During this review, it was observed that certain papers did not consider the bond spread as the dependent variable. Instead, they used the bond spread to predict other variables such as GDP. Moreover, it should be noted that some of the papers we reviewed did not carry out the out-of-sample forecasting that aligned with our interests. Following this evaluation, we selected the most relevant papers for our in-depth literature review, resulting in the final selection of 20 papers shown in Table 1.

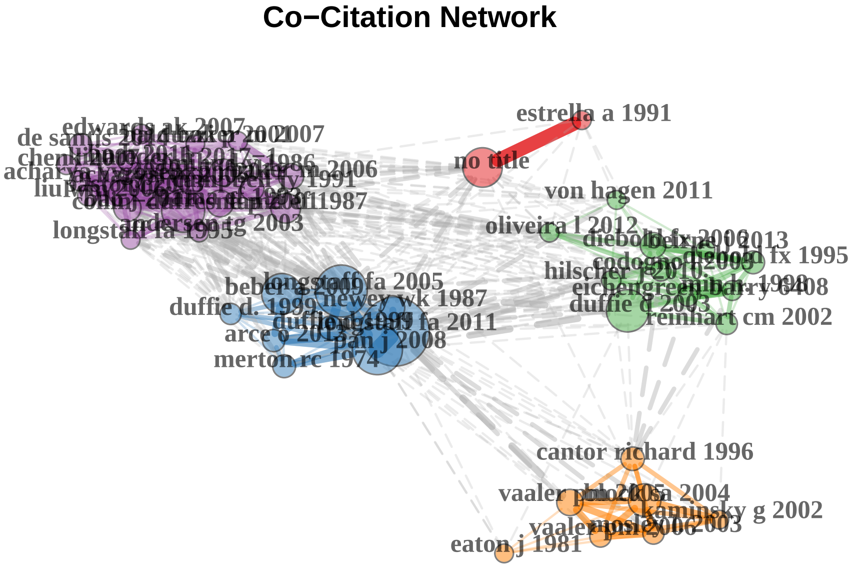


Figure 8: Co-citation network plot. The figure shows the 5 clusters of articles that were identified by the Louvain algorithm. The network is constructed based on the 614 papers identified from the search for ”sovereign bond spread*” on WoS.

B Data Extraction

FRED

The data from FRED was collected to a DataFrame in Python by using the FRED API (FRED Python Development Team, 2021). The variables collected from FRED are shown in Table 20.

Variable	**Type**	**Ticker**
Real GDP	Country Specific	CLVMNACSCAB1GQDE
Cash surplus/deficit	Country Specific	CASHBLDEA188A
General government gross debt	Country Specific	GGGDTADEA188N
CPI all items total	Country Specific	CPALTT01DEM659N
CPI all items non-food non-energy	Country Specific	CPGRLE01DEM659N
GFCF	Country Specific	DEUGFCFQDSMEI
Export of goods and services	Country Specific	DEUEXPORTQDSNAQ
Price level of exports	Country Specific	PLXCPPDEA670NRUG
Price level of imports	Country Specific	PLMCPPDEA670NRUG
Fed Interest Rate	Global Conditions	DFF
Stock Market IV	Global Conditions	EMVMACROBUS
BBB-rated US corp. bond yield	Global Conditions	BAMLC0A4CBBBEY
Gov final consumption expenditure	Country Specific	DEUGFCEQDSNAQ
Gross national income	Country Specific	MKTGNIDEA646NWDB
Prime lending rate	Global Conditions	DPRIME
US stock market return	Global Conditions	SP500
Oil price change	Global Conditions	POILBREUSDM
Gov 10y to 90-day bill US	Global Conditions	T10Y3M
Unemployment	Country Specific	LRHUTTTTDEM156S
Local interbank rate	Eurozone Conditions	IR3TIB01DEM156N
Volatility Index (VIX)	Global Conditions	VIXCLS
General gov. debt (% GDP)	Country Specific	GGGDTADEA188N

Table 20: The data collected from FRED. The German Ticker is presented for the country-specific variables, whereas the ticker can be found under the corresponding names for other countries.

Refinitiv Eikon

To collect data from Refinitiv Eikon and insert it into a DataFrame, the Eikon Python API wrapper by Ankile and Krange (2022) was applied. The API wrapper provides a more accessible and user-friendly way of retrieving financial data than using the Python package provided by Eikon directly. The variables collected from Eikon are shown in Table 21.

Heritage

Heritage was used as a source of freedom indicators and political data. Table 22 shows the variables collected from Heritage. The data from Heritage is found under the variable name and consequently has no ticker. The data was downloaded as a CSV file from The Heritage Foundation (2023) and imported to Python as a DataFrame.

Variable	**Type**	**Ticker**
Current account balance	Country Specific	aDECURAC
Import of goods and services	Country Specific	aDEIMP/CA
Bond yield from US treasury 1y	Global Conditions	US1YT=RR
GDP per capita	Country Specific	aDECGDHD/CA
GFCF	Country Specific	aDEGFCF/CA
Return on regional stock index	Country Specific	.GDAXI
EU Policy rate	Global Conditions	EUECBR=ECI
Yield 10Y Sovereign Bond	Country Specific	DE10YT=RR

Table 21: The data collected from Refinitiv Eikon. The German Ticker is presented for the country-specific variables, whereas the ticker can be found under the corresponding names for other countries.

Variable	**Type**
Property Rights	Country Specific
Judicial Effectiveness	Country Specific
Government Integrity	Country Specific
Tax Burden	Country Specific
Government Spending	Country Specific
Fiscal Health	Country Specific
Business Freedom	Country Specific
Labor Freedom	Country Specific
Monetary Freedom	Country Specific
Trade Freedom	Country Specific
Investment Freedom	Country Specific
Financial Freedom	Country Specific

Table 22: The data collected from Heritage. The data was available annually and required interpolation for this use case.

C ANN

Background

Warren McCulloch and Walter Pitts introduced a simplified model of a biological neuron in 1943. It is today known as the McCulloch-Pitts (M-P) neuron (McCulloch and Pitts, 1943). Extending upon this, the perceptron was developed in the late 1950s by Frank Rosenblatt (Rosenblatt, 1958). As computing power was low at the time, the excitement for the perceptron faded. However, in the 1970s and 80s, increased computing power and the discovery of the backpropagation algorithm restored interest in ANNs. Backpropagation enabled the development of complex networks by combining perceptrons, leading to the popularization of MLPs. Today, we witness the emergence of deeper networks with an increased number of perceptrons, along with the introduction of new network structures.

The Perceptron

The perceptrons can be seen as the building blocks of ANNs and are often referred to as neurons within the context of ANNs. A perceptron has an input layer, a bias, a weight vector related to the input and bias, a summation function for the inputs, an activation function, and an output. The input to a perceptron is a vector of numbers. These numbers can originate from specific features, which in our case are economic variables.

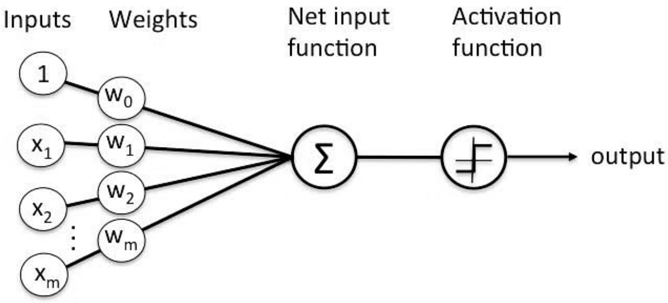


Figure 9: The perceptron consists of an input layer, weights, a bias, a summation function, an activation function, and an output. The inputs and bias are multiplied by the associated weight, summed up, and passed through the activation function before being passed on as an output. The input of "1" represents the bias in this figure.

Each of the inputs is associated with a weight in the weight vector. The weight determines the importance of the feature input in the classification or regression process. The weight is multiplied with the input and all the weighted inputs are summed up with a summation function. For example, let's consider a scenario where one of the inputs into the perceptron is GDP_CAPITA with a value of 100. If the weight associated with this input feature is 0.5, then the "effect" of GDP_CAPITA on the output would be 50.

The bias serves as a constant input with an adjustable weight in the perceptron. It is added to the rest of the inputs, providing flexibility in adjusting the activation threshold. The dot product of the input vector and weight vector, including the bias, is passed through an activation function. Typically, the activation function is a threshold function. When the input exceeds a certain threshold, the perceptron activates and produces an output associated with that activation.

To learn, the perceptron adjusts its weight vector to optimize the performance. To do this, one often uses the perceptron learning rule. The perceptron computes the output based on a given input, compares it to the true output value, and calculates the error. To adjust the weights, the following equation is used:

$$\Delta w_i = \eta \cdot (target - predicted) \cdot input_i \tag{11}$$

The output of the formula, Δw_i, is the change in weight i associated with input feature number i. The change in weight i is related to the size of the error, a constant learning rate η, and the size of input i. This learning process is done for a given number of epochs on the dataset.

The Activation Function

The activation function is the core operation in a perceptron. The perceptron has important features for effective learning, such as non-linearity to grasp complex data relationships and monotonicity to ensure input-output consistency. In addition to the standard threshold function, which outputs 1 above a certain threshold and 0 below it, there are various activation functions available to choose from. Three of the functions we have explored are ReLU, ELU, and Tanh.

According to the findings presented in Parhi and Nowak (2020), the preferred activation function in neural networks is ReLU. Hence, our hypothesis was that the ReLU activation function would give the best performance. The ReLU function sets all inputs below 0 to 0 and has a linear output above 0, as seen in Figure 10. The ReLU function was developed to increase data sparsity by decreasing the number of active neurons (Parhi and Nowak, 2020). Therefore, training processes of neural networks using ReLU tend to converge, and the computational cost of the training processes is low, resulting in faster training. However, ReLU is prone to the "dying ReLU" issue. The problem comes from the fact that negative values are discarded as they are set to zero. This results in a derivative of zero for negative values, and no weight update during backpropagation (Biewald, 2020).

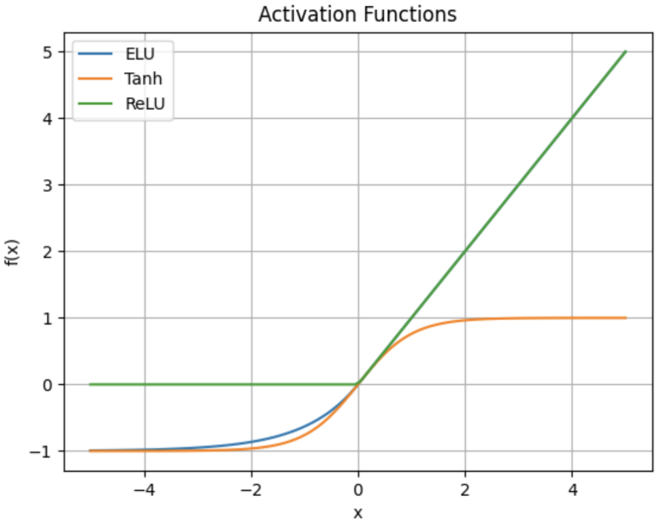


Figure 10: Visualization of the three activation functions ReLU, ELU and Tanh

Clevert et al. (2015) introduced the ELU as a solution to the "dying ReLU" issue. ELU also has a linear output for input values above 0 but transforms input values below 0 to $\alpha(exp(x) - 1)$. ELU permits negative output, which helps to bring the average unit activation closer to zero. Another alternative is the Tanh function. However, it also suffers from the vanishing gradient problem observed in the ReLU function. Goodfellow et al. (2016) recommend the use of the Tanh function in the hidden layers of ANNs for regression tasks. The well-known mathematical Tanh function transforms both positive and negative values, ensuring that the output remains within the range of -1 to 1. This characteristic helps in normalizing the data.

Multilayer Neural Networks

A multilayer neural network, also known as a feedforward neural network or an MLP, is a type of ANN that consists of multiple layers of neurons. Multilayer neural networks are widely used and known for their ability to learn complex patterns to make predictions. The network consists of connected perceptrons that receive the input from the previous layer and pass their output to the next layer. This propagation through the network is called feedforward. The network usually consists of an input layer, one or more hidden layers, and an output layer.

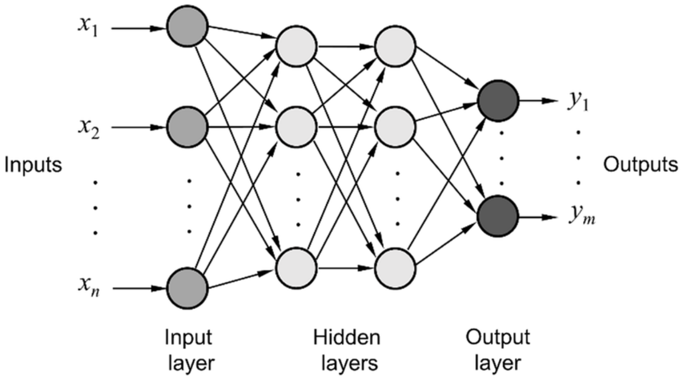


Figure 11: Visualization of a multilayer artificial neural network. These network structures have an input layer, one or more hidden layers, and an output layer. The layers consist of multiple perceptrons, often referred to as neurons.

By adding hidden layers to the network, the network can progressively learn more complex patterns from the input data. The output layer of the network produces the final prediction or decision based on the learned representations from the hidden layers. The number of neurons in the output layer depends on the type of problem the network is designed to solve. For instance, in a regression task, the output layer can consist of a single neuron with a linear activation function, that produces a regression prediction.

To train a multilayer neural network, a process known as backpropagation was used. Backpropagation involves computing the gradients of the network's parameters (weights and biases) with respect to a given loss function. The gradients are used to update the parameters using an optimization algorithm known as gradient descent. The objective is to minimize the difference between the network's predictions and the true values.

D LightGBM

Background
LightGBM is a gradient-boosting framework (Ke et al., 2017). Boosting is based on the idea that a weak learner can be progressively modified to improve. The idea of an adjustable weak learner can be traced all the way back to Kearns (1988), which referred to what we call learners as ”hypotheses”. A weak learner is a prediction algorithm, or more specifically in the case of LightGBM, a tree structure. Hypothesis boosting, as it was called, proposed the use of a sequence of weak learners. Each new weak learner would focus on the instances that were misclassified by the previous weak learner. Therefore, you would have a group of weak learners that can classify cases together. By adding new weak learners, they can progressively adapt and learn from new datasets.

Adaptive Boosting (Freund and Schapire, 1995), or AdaBoost, was the first successful implementation of boosting. AdaBoost uses decision trees with a single split, called decision stumps. AdaBoost adds new stumps that focused on the more difficult patterns to predict until the algorithm deems the pattern sufficiently solved. The predictions are made by a vote among the stumps, where the votes are weighed by the individual stumps' accuracy.

Later, Friedman (2001) introduced a framework called Gradient Boosting Machines, now often referred to as gradient boosting. As the name suggests, this framework uses a gradient descent procedure to minimize the loss of the overall model when adding new learners. In contrast to earlier models, the existing learners are frozen when a new learner is introduced. As this new method allowed for an arbitrary loss function, boosting was no longer just a binary classification framework. For example, it could now support multiclass classification and regression.

The LightGBM Framework
LightGBM is a boosting framework using leaf-wise growth for its weak learners, as seen in Figure 12. It offers three types of gradient boosting methods that can be selected by specifying the boosting parameter: GBDT, DART, and GOSS.

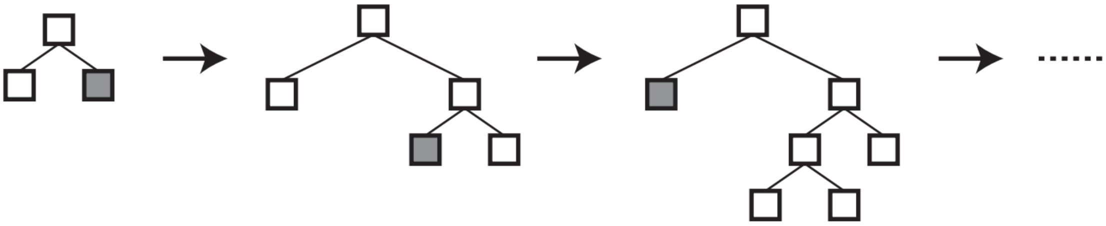


Figure 12: Visualization of leaf-wise growth in LightGBM. Adapted from (Quinto, 2020, p. 347).

Friedman (2001) originally proposed the default GBDT method. GBDT is the most well-known method and is known to be reliable and stable, but also prone to overfitting and being memory intensive. This results in the efficiency being unsatisfactory for large data sets. Ke et al. (2017) introduced LightGBM, with GOSS, to tackle this challenge.

GOSS is a gradient boosting method that focuses on samples considered challenging and only gives attention to *some* of the easier samples. The way GOSS determines how challenging samples are is by evaluating their gradient. Large gradients are more challenging than smaller gradients. By excluding lower gradient data instances, the model's efficiency increases, in addition to improved generalization in the model. Furthermore, by selectively discarding less informative instances, the model focuses on more meaningful patterns in the data, which also prevents it from memorizing noise or outliers.

The DART method is an enhancement of the traditional Multiple Additive Regression Trees (MART) method that addresses the issue of over-specialization in traditional Gradient Boosted Trees. When trees are added in later iterations, they impact only a few instances and make vanishing contributions to the rest. However, the model becomes oversensitive to initially added trees instead of generalizing smoothly across the ensemble. To mitigate this effect, Vinayak and Gilad-Bachrach (2015) introduced dropout to the MART model, resulting in the DART model, which involves randomly dropping trees and can be seen as a form of regularization.

E Autocorrelation Plots

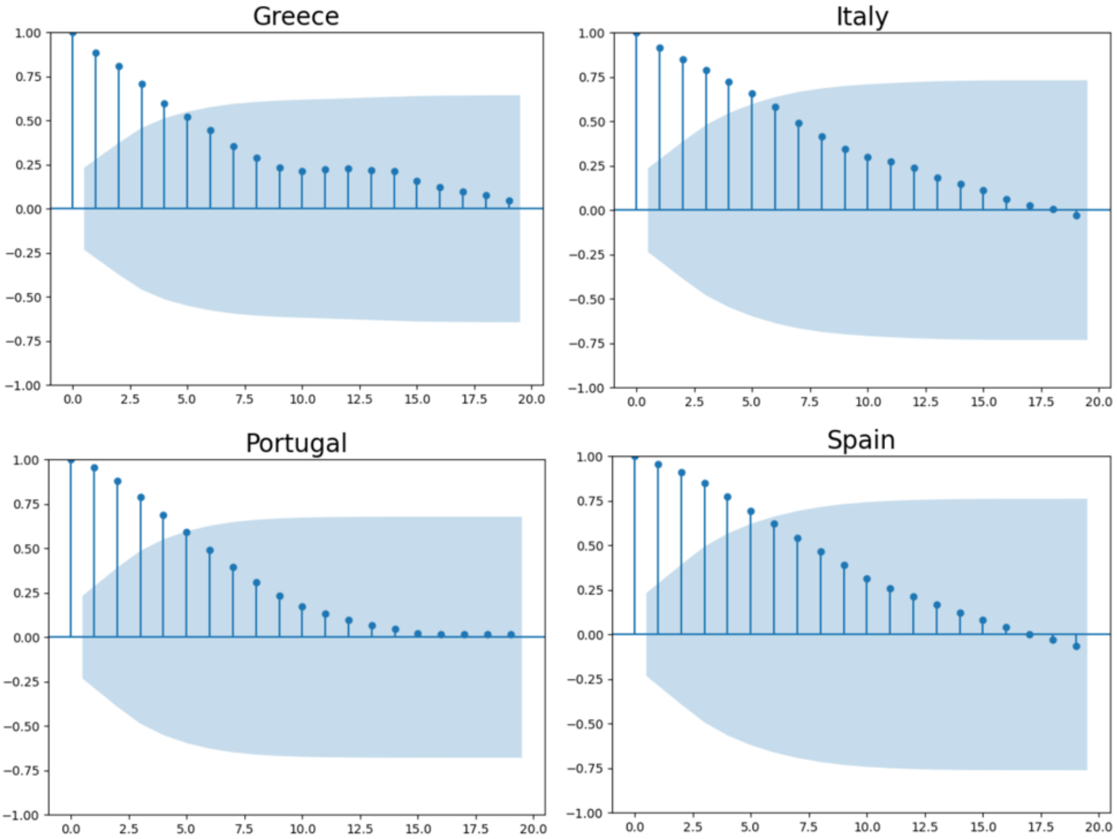


Figure 13: The figure displays the auto-correlation functions of the bond spreads. The plot shows five significant spikes for Portugal and Greece and six significant spikes for Italy and Spain.

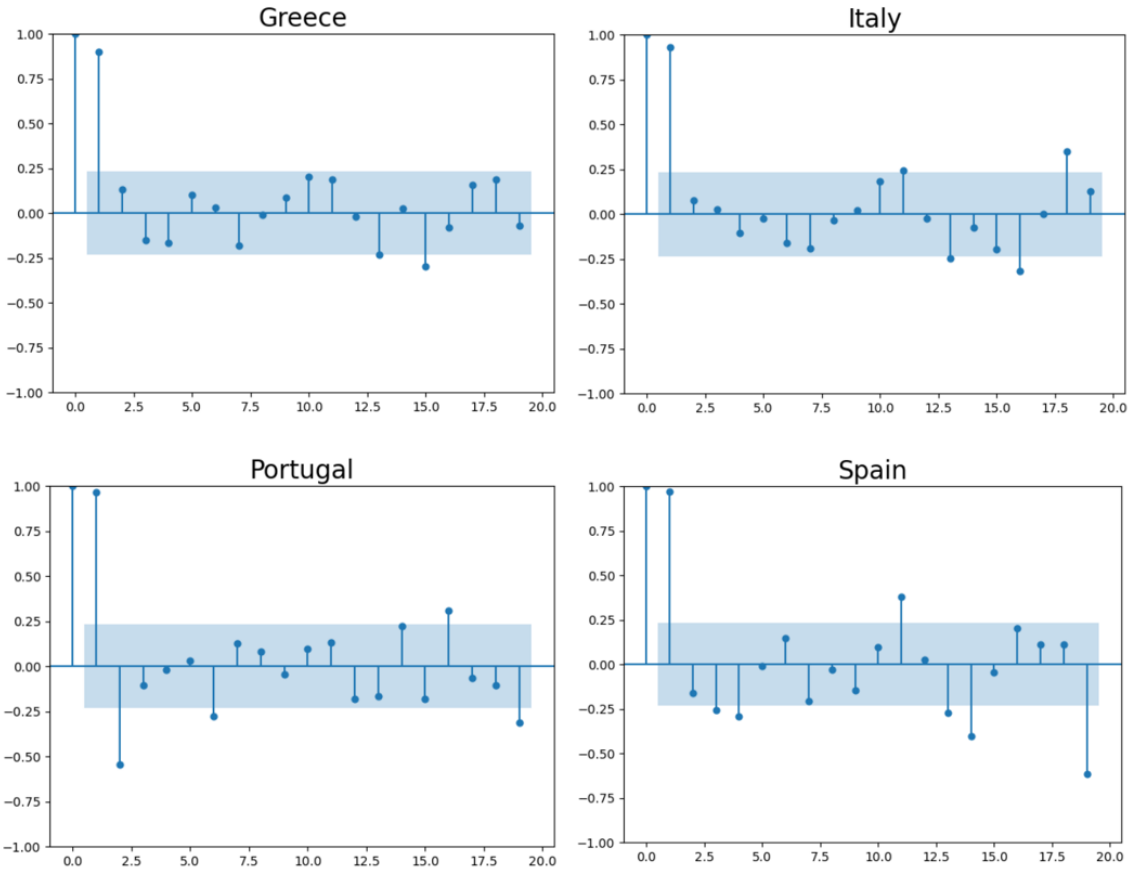


Figure 14: The figure shows the partial auto-correlation functions of the bond spreads. The plot displays three significant spikes for Portugal and two significant spikes for the remaining three countries.

F WandB Hyperparameter Tuning

WandB is a platform designed to help researchers track, visualize, and manage projects in a collaborative manner (Biewald, 2020). It offers a range of features to streamline the machine-learning workflow and facilitate experimentation. WandB provides a centralized dashboard that allows users to log and monitor various aspects of their experiments. It integrates with libraries such as Keras, enabling easy logging of metrics, hyperparameters, and visualizations.

One of the key features of WandB is the concept of "sweeps." A sweep refers to an automated hyperparameter search process that explores different combinations of hyperparameters to find the optimal configuration for a model. Sweeps are particularly useful for hyperparameter tuning, as they help search through a wide range of possible values and identify the combination that yields the highest performance. In a WandB sweep, one can specify what hyperparameters to tune and define their respective search spaces. The sweep automatically generates different configurations by sampling hyperparameter values from the specified search spaces. Each configuration is associated with a unique set of hyperparameters, and a test is run using that configuration.

For a sweep, one can also define the manner in which the search space is explored. There are mainly three ways of exploring the search space; random search, grid search, and Bayesian search. Random search randomly explores the search space, while grid search methodically works through all possible combinations. The Bayesian search uses Bayesian inference to explore better and exploit the search space. Bayesian search is well suited for models with expensive computations, such as a ANN (Snoek et al., 2012). We thus used Bayesian search for our ANNs and random search for our LightGBMs.

During the sweep, WandB tracks and logs each experiment's performance metrics, hyperparameter values, and other relevant information. This allowed us to visualize and compare the results across different configurations, and make informed decisions about the optimal configuration. Figure 15 illustrates our hyperparameter search for the ANN, while Figure 16 visualizes the hyperparameter search for the LightGBM.

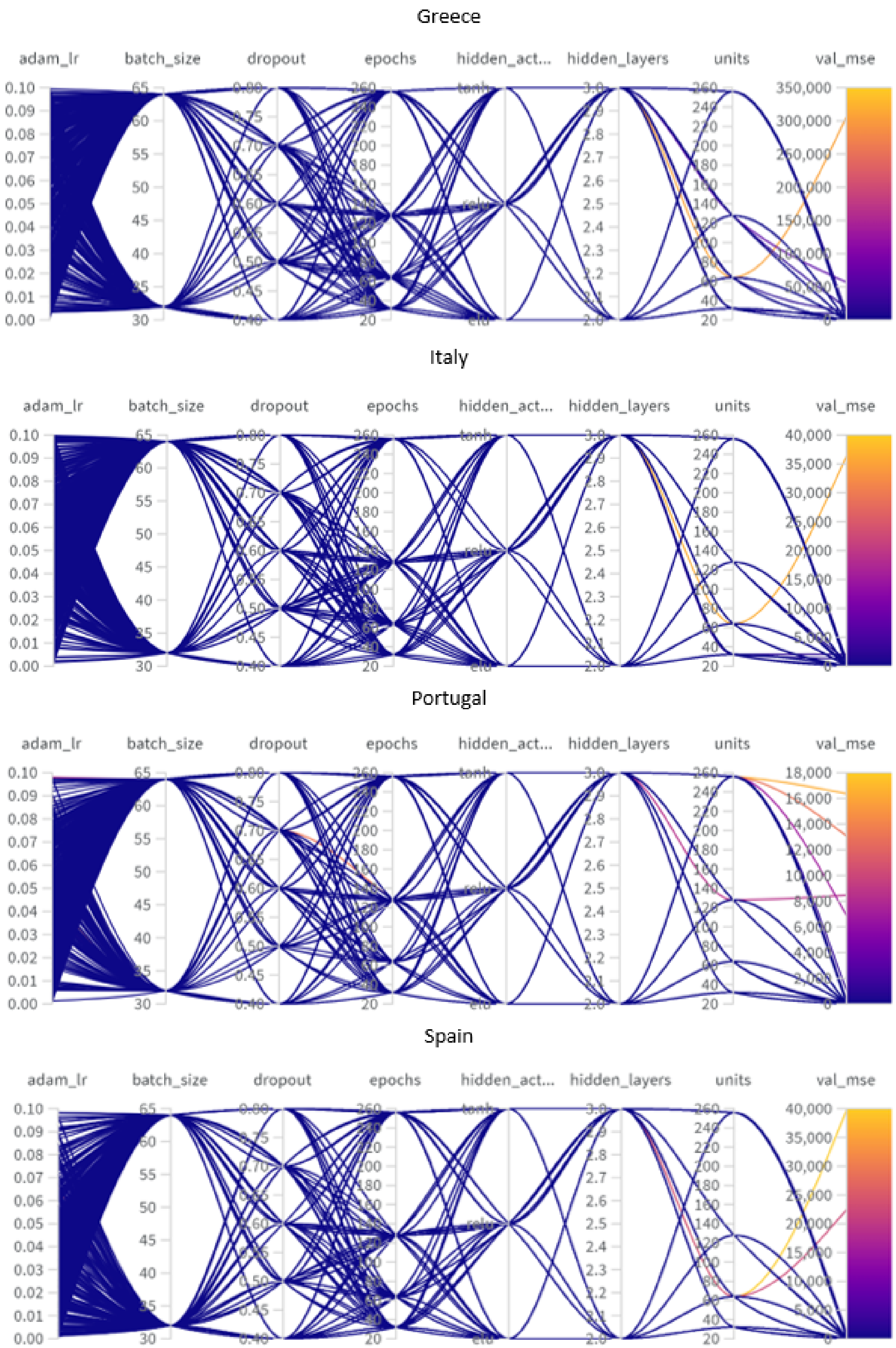


Figure 15: Hyperparameter search for ANN by country. Different combinations of hyperparameters were tested through 500 runs. The configuration of hyperparameters that yielded the lowest MSE on the validation set (indicated as `val_mse` on the right) were chosen as the hyperparameters for the models.

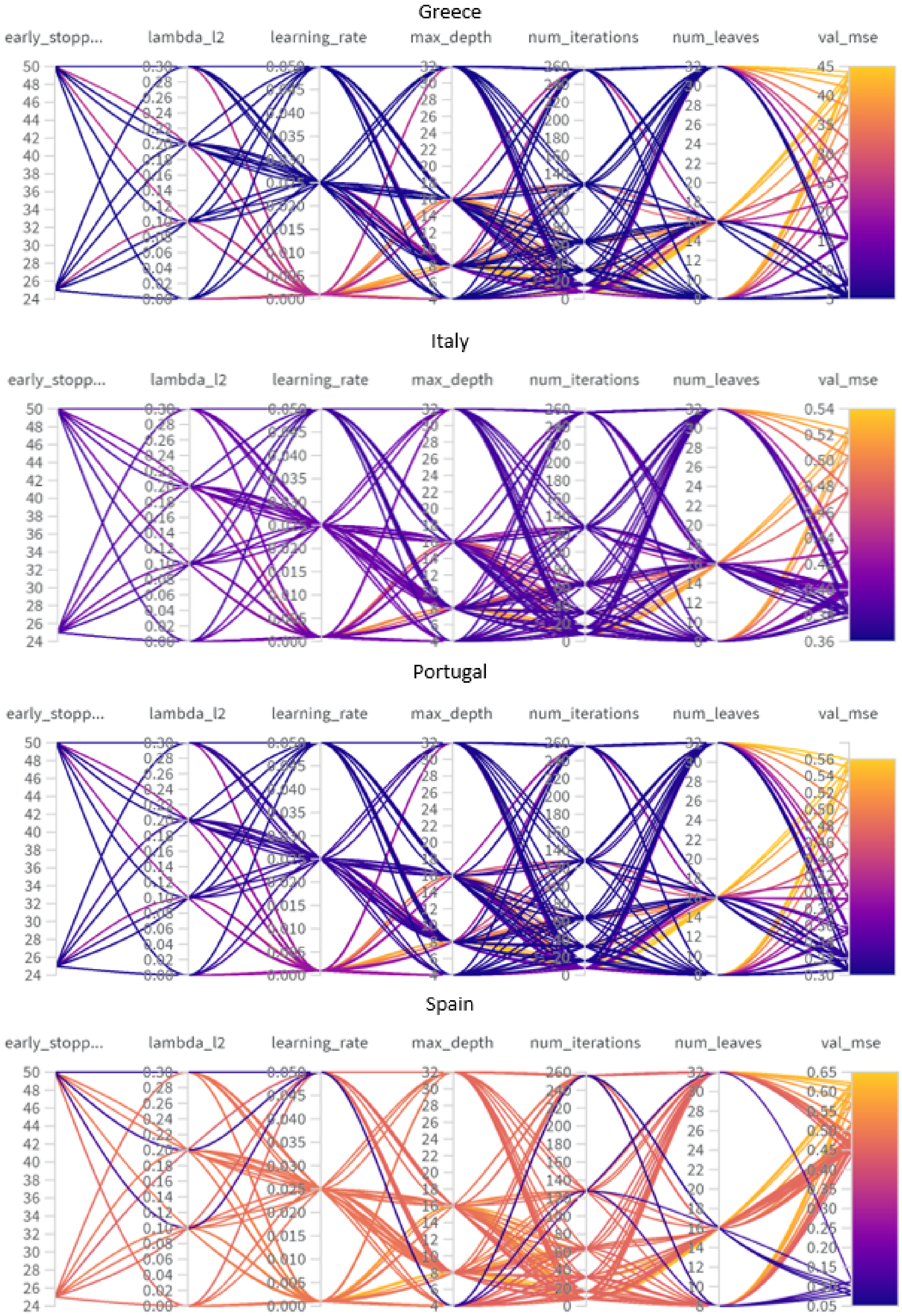
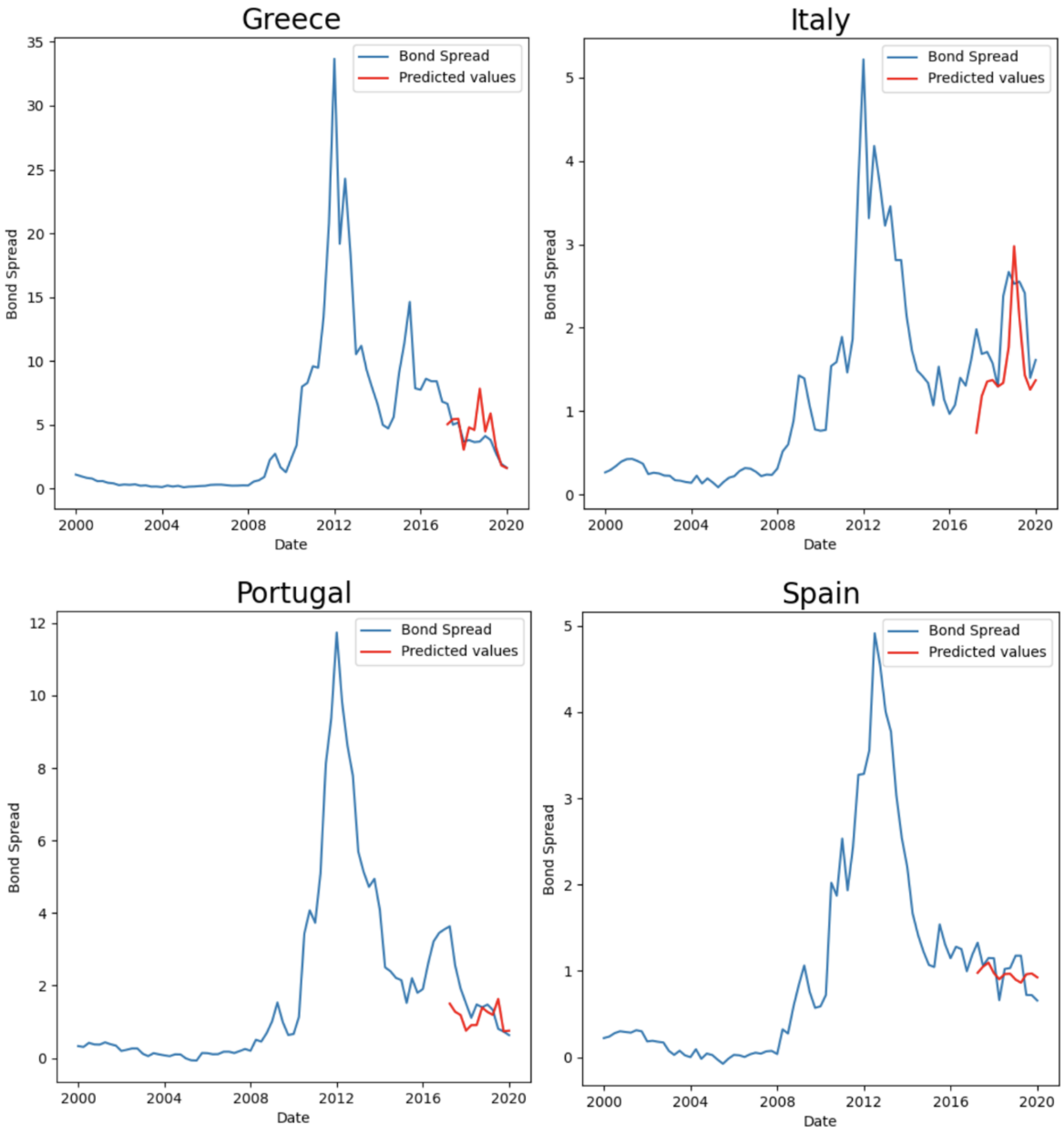


Figure 16: Hyperparameter search for LightGBM by country. Different combinations of hyperparameters were tested through 500 runs. The configuration of hyperparameters that yielded the lowest MSE on the validation set (indicated as `val_mse` on the right) were chosen as the hyperparameters for the models.

G Model Prediction Plots



Greece

Italy

Portugal

Spain

Figure 17: The figure shows the predicted values of the ANN model in contrast to the actual values, categorized by the four countries. Overall, the model demonstrates capability in predicting upward or downward movements. However, it encounters challenges in accurately estimating the magnitude of the moves.

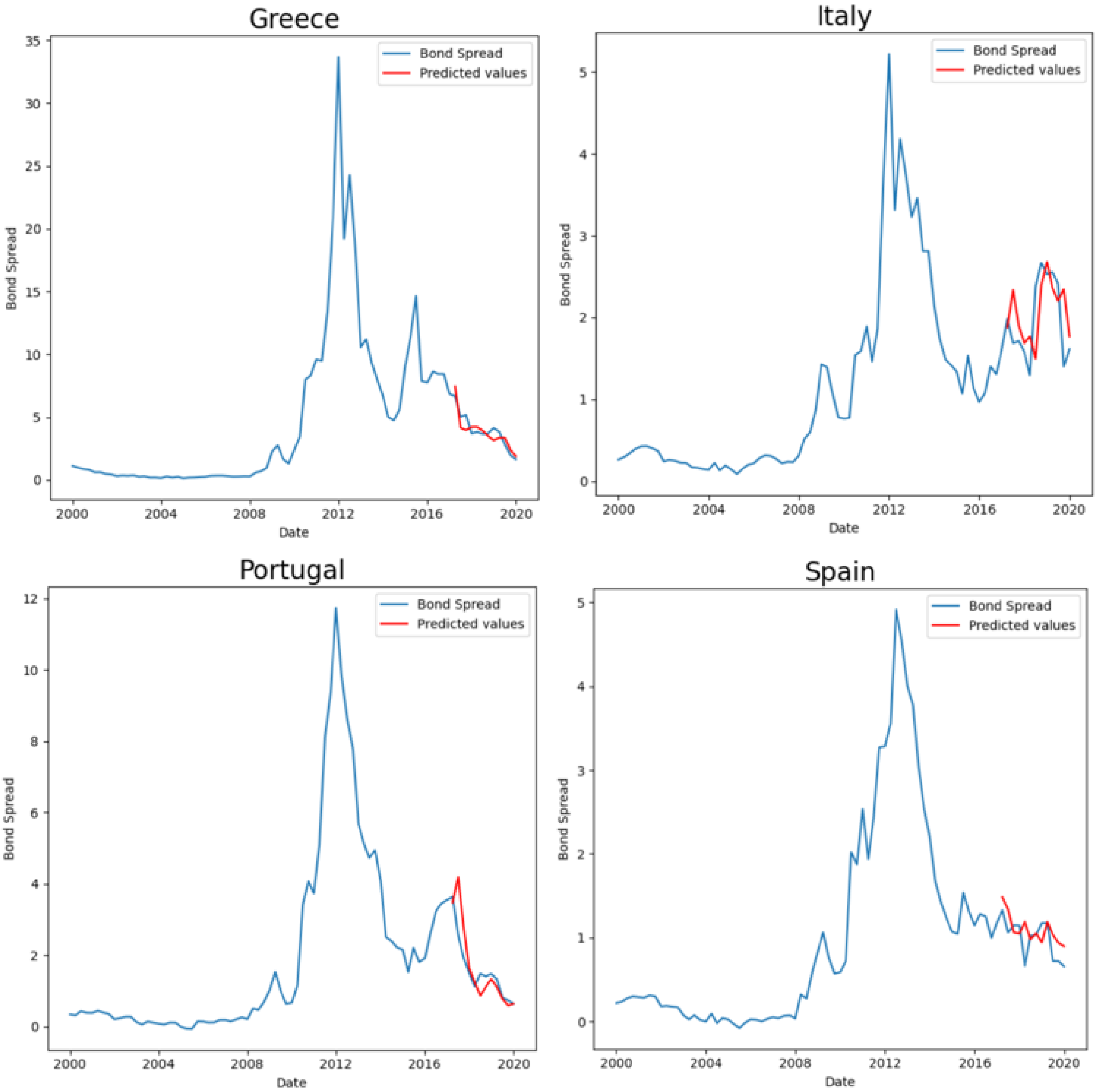


Figure 18: The figure displays the predicted values of the LightGBM model in comparison to the actual values, categorized by the four countries. Overall, the model demonstrates effective prediction capabilities, but with a few notable instances of larger errors.

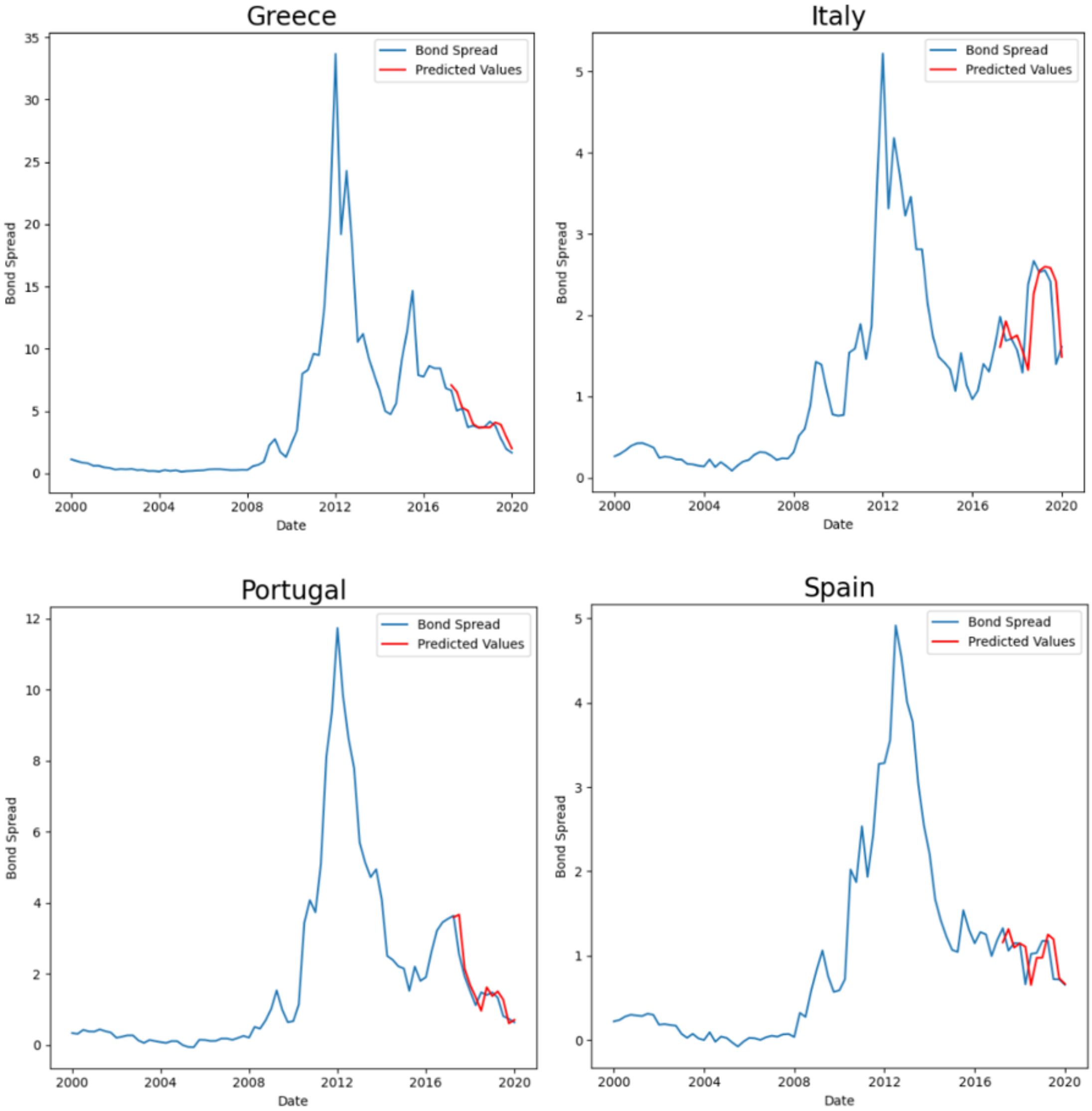


Figure 19: The figure illustrates the AR model's predicted values compared to the actual values, categorized by the four countries. The model's predictions are frequently very close to the previous value, resulting in a lagged pattern in some instances in the figure.

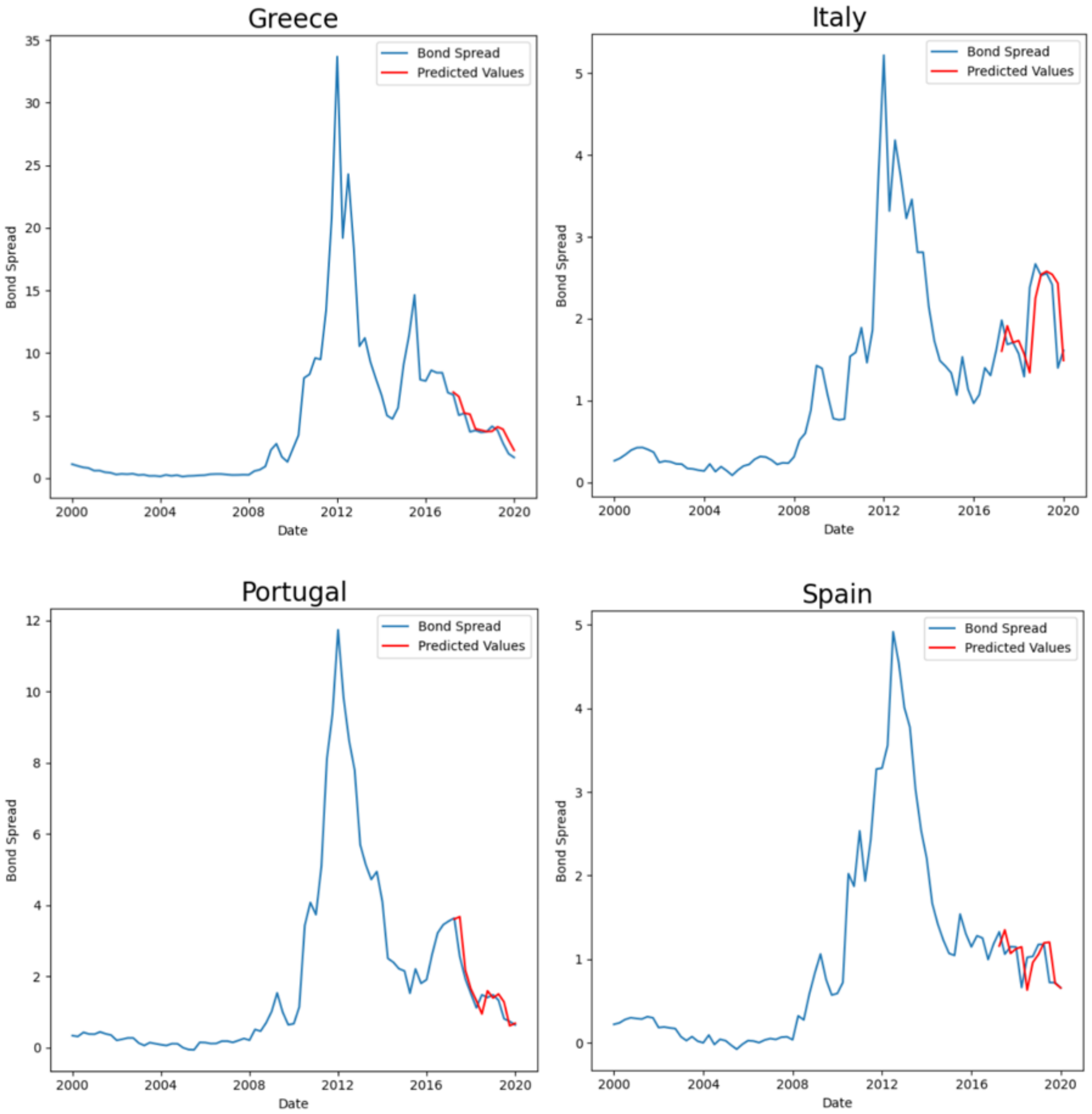


Figure 20: The figure illustrates the ARIMA model's predicted values compared to the actual values, categorized by the four countries.

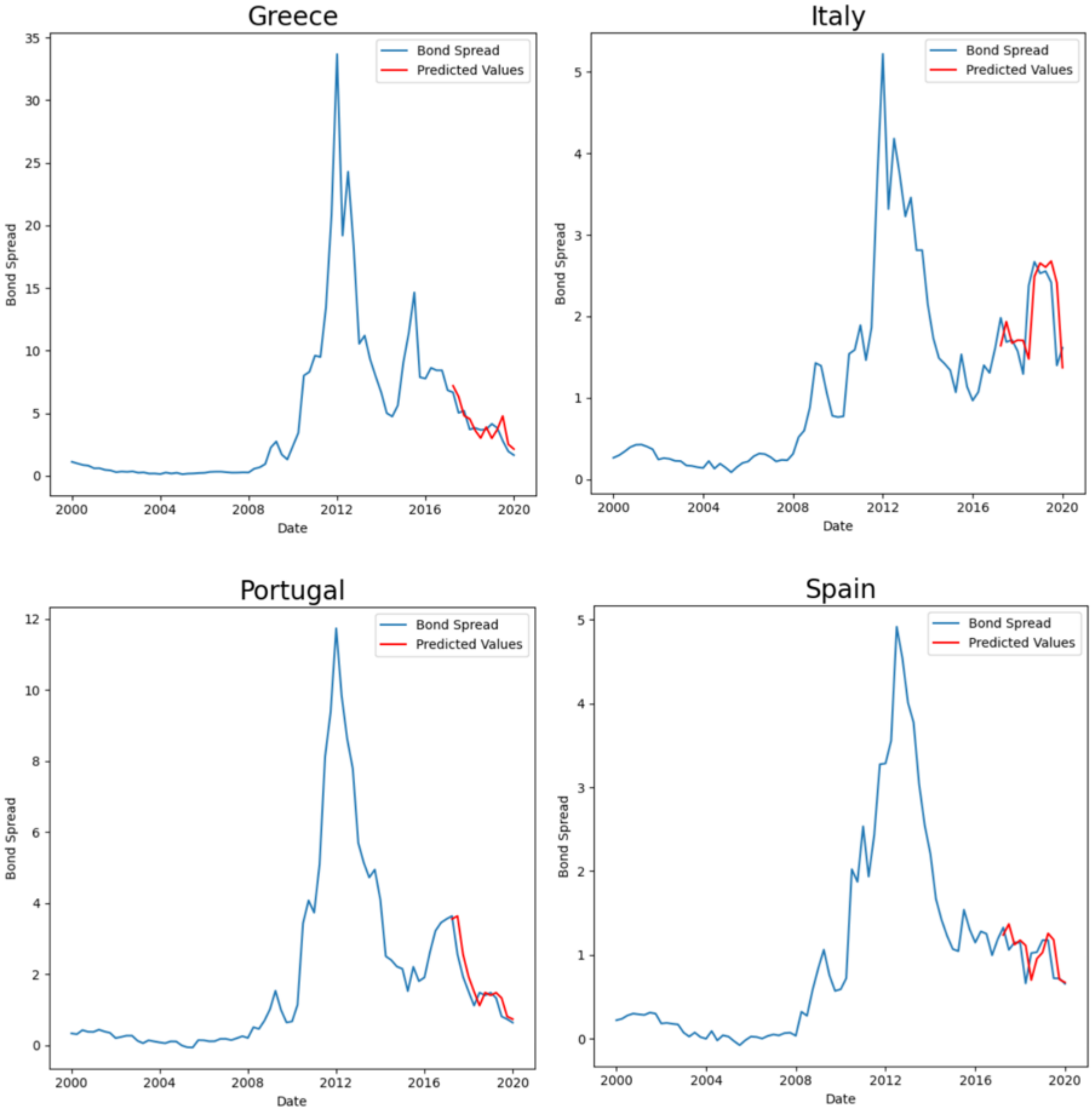


Figure 21: The figure illustrates the ARIMAX model's predicted values compared to the actual values, categorized by the four countries.

H ARIMAX Model Coefficients

Data	Greece	Italy	Portugal	Spain
GDP_CAPITA	0.003	0.000	-0.000	0.000
GDP_GROWTH	0.305	0.000	-0.196	0.010
GFCF	-0.016	-0.087	0.066	-0.033
EXPORT	-0.031	-0.008	-0.079	-0.083
CPI_TOT	-0.834	-0.679	-0.656	-0.065
CPI_NON_FOOD	1.138	0.588	0.507	0.058
STOCK_INDEX	-0.546	0.020	0.258	0.309
UNEMPLOYMENT	0.747	0.335	-0.124	0.279
GOVDEBT	0.269	-0.233	-0.312	-0.032
GOVINT	0.156	-0.107	0.307	-0.014
GOVSPE	-0.252	-0.006	-0.450	-0.012
MONFRE	-0.866	-0.756	0.738	0.033
INVFRE	0.204	-0.029	-0.118	0.014
FINFRE	-0.013	0.099	-0.043	0.010
DFF	-0.074	-0.085	-0.067	0.066
BBB_yield_US	-0.733	-0.130	-0.001	-0.126
SP500	0.027	-0.004	0.113	-0.006
VIX	0.166	0.021	-0.002	-0.012
ECB_INTERBANK_RATE	1.220	0.023	-0.214	-0.041
DE_GDP_CAPITA	0.003	0.000	-0.001	0.000
DE_IMPORT	0.112	0.065	0.099	0.101
DE_CPI_TOT	1.019	0.028	0.167	0.202
DE_UNEMPLOYMENT	2.995	0.547	-1.380	0.331
DE_GOVDEBT	0.385	0.113	0.830	-0.027
DE_PRORIG	0.604	-0.367	-0.964	-0.013
DE_GOVSPE	0.124	-0.039	-0.545	-0.036
10YT_SPREAD_LAG1	-1.053	-0.120	0.000	0.587
10YT_SPREAD_LAG2	-0.205	-0.309	0.000	0.159
MA_LAG1	0.802	-0.025	0.000	-0.664
SIGMA2	5.231	0.106	0.240	0.071

Table 23: The table displays the coefficients associated with the variables in the ARIMAX model. These coefficients exhibit substantial variations in both sign and magnitude when comparing across the different countries. Sigma2 represents the variance of the error term that captures the unexplained variation.

NTNU

Norwegian University of
Science and Technology