A Quantitative Study on the Utilization of ESG Financial Market Performance Metrics in Forecasting Finland's Economic Trajectory

Aleksi Vuorio

Haaga-Helia University of Applied Sciences

Bachelor of Business Administration
Bachelor’s Thesis
2025

Abstract

Author Aleksi Vuorio	
Degree Bachelor of Business Administration	
Thesis Title A Quantitative Study on the Utilization of ESG Financial Market Performance Metrics in Forecasting Finland's Economic Trajectory	
Number of pages and appendix pages 47 + 58	

This thesis investigates the utility of ESG (Environmental, Social, and Governance) financial market performance metrics in forecasting economic trajectory, particularly that of Finland's. Inherently shaped by The Nordic Welfare Model, Finland's economy has endured persistent stagnation post-2008 financial crisis, attributed to the culmination of ageing population, static labour productivity and inadequate investment. In light of the emergence of ESG-oriented finance in recent years, this study seeks to assess the potential predictive relationships between ESG financial metrics and key macroeconomic indicators, specifically GDP, inflation and employment. This analysis is supplemented by a comparison to traditional financial market performance, and five-year economic forecasts generated using two models incorporating ESG metrics. The scope of the study consists of quarterly data from Q1 2011 to Q4 2024, focusing on domestic aggregate macro data, which excludes sector-level and pillar-specific ESG data. The scope also excludes any aim to prove causality, rather focusing on correlation, and explicitly accounting for structural breaks in the time series, treating the temporal sequence as a monolith.

The research rationale is reinforced by Stakeholder Theory (Freeman 1984) and Creating Shared Value (Porter & Kramer 2011), which forms the cornerstones of the theoretical framework. The framework is further complemented by relevant academic research and a contextual link to the mechanisms of The Nordic Welfare Model, allowing for a rational formulation of a hypothesis that Finnish ESG-screened firms exhibiting strong financial performance would correlate with thriving economic growth, and vice-versa. The variable chosen to represent those firms is a gross index OMXSUSTAINFIGI, which represents Finnish companies screened for defined ESG criteria. The variable was controlled with a global ESG index and compared to a Helsinki Stock Exchange all-share index OMXHGI. The empirical analysis was conducted in Python through econometric methods. The main tool applied was the Vector Error Correction Model (VECM) to test both short-term dynamics and long-run equilibrium relationships. Stationarity was tested using the Augmented Dickey-Fuller test, and cointegration was assessed via the Johansen method. Model accuracy was evaluated using RMSE and MAE.

The results suggest that ESG performance demonstrates a moderately to significantly positive link to GDP and employment in the short run and to GDP and inflation in the long run, though short-term correlations diminish once controlling for global influences. ESG-based models performed competitively with traditional benchmarks, particularly in forecasting GDP and inflation, though they underperformed in predicting employment. Employment was the only variable found to adjust significantly to restore long-run equilibrium. Forecasts project modest economic recovery over the next half-decade, though all models exhibited systematic forecast biases. It was concluded that ESG financial performance holds value as a complementary tool in macroeconomic forecasting, particularly in sustainability-oriented economies such as Finland. However, limitations such as potential structural breaks in the time series and the requisite for disaggregated ESG data were noted as areas for further research.

Key words
ESG Metrics, Economic Forecasting, Sustainable Finance, Econometric Analysis, Finnish Welfare Economy, Macroeconomic Indicators

Table of Contents

1 Introduction

This introduction chapter lays the foundation for the thesis by providing essential background and framing the research focus. It begins with a General Introduction outlining Finland’s post-2008 economic stagnation and the global rise of ESG finance. Following this, the Research Problem and Objectives are presented, culminating in a clear research question and supporting investigative questions. The Limitations and Scope section defines the study’s boundaries, while Key Concepts explains crucial terms and frameworks that will be used throughout the research.

1.1 General Introduction

Finland’s economy has experienced a prolonged period of sluggish growth and structural challenges in the post-2008 era. The nation’s economic stagnation is most evidenced by persistently low GDP growth and stagnant GDP per capita growth since the global financial crisis. (Kokkinen, Jalasjoki & Obstbaum 2025.) Aggregate productivity growth also effectively stagnated between 2008 and 2016, only partially rebounding in subsequent years (Calligaris, Jurvanen, Lassi, Manaresi & Verlhac 2023, 5).

In late 2024, Finland was projected to be the worst-performing economy in the euro area, following a second consecutive year of contraction (Etla 2024). This marked the culmination of long-standing structural weaknesses, including an ageing population, insufficient investment in human capital and R&D, and stagnant labor productivity (Kokkinen et al., 2025). The collapse of key industries, sluggish domestic demand, and slow adjustment to post-pandemic and geopolitical shifts further deepened the downturn.

Although the economy started to recover gradually in mid-2024, overall growth remains weak. In 2025, GDP is expected to increase by only 1.2 %, with net exports continuing to hinder the recovery (Etla 2025). Without major structural reforms, long-term forecasts suggest GDP growth will average only 1.8 % annually in the 2030s (Kokkinen et al. 2025). Finland, previously regarded as a resilient Nordic economy, now confronts persistent structural challenges, including subdued growth, increasing fiscal constraints, and macroeconomic uncertainty.

Concurrently, ESG (Environmental, Social, and Governance) finance has seen a global surge, reflecting investors' increasing focus on sustainability criteria in capital markets. What was once a niche area has become mainstream over the past decade (Orlando 2024). In 2022, global sustainable investment assets totaled around $30 trillion and are expected to grow significantly (Global Sustainable Investment Alliance 2023, 5). According to Bloomberg Intelligence, the ESG market is projected to exceed $40 trillion by 2030, growing at a compound annual rate of 3.5 % and accounting for more than 25 % of the anticipated $140 trillion in assets under management (Diab & Mahtani 2024).

This substantial growth in ESG finance suggests an increasing integration of sustainability and governance considerations into capital market assessments of corporate performance. As ESG funds and indices attract record inflows, they may also become indicators of broader economic shifts, especially as economies transition toward sustainability goals (e.g. low-carbon technology, social inclusion, good governance). Finland, known for its advanced welfare state and high education and environmental standards, is posited to both contribute to and be influenced by these ESG trends. This study examines whether ESG financial market performance contains information relevant to forecasting Finland's macroeconomic development. Given the persistent stagnation in Finland's economy, the identification of forward-looking indicators is of analytical interest. If strong ESG performance aligns with positive economic outcomes, ESG metrics could serve as a barometer for Finland's future growth.

1.2 Research Problem and Objectives

Against a backdrop of prolonged stagnation in Finland's economy and the rapid mainstreaming of ESG finance, this thesis explores whether ESG financial performance, specifically ESG market indices, can serve as a meaningful tool in forecasting economic growth in Finland. By examining the relationship between ESG metrics and macroeconomic outcomes, the study aims to assess the potential of sustainable finance as a complementary lens in national economic forecasting. While ESG-related research has grown globally, there is a notable lack of studies focused on Finland, leaving a gap this thesis seeks to address.

1.2.1 Research Question

Can ESG financial market performance be utilized effectively to predict economic growth in Finland?

1.2.2 Investigative Questions

1. How does Finland's ESG market index correlate with key macroeconomic indicators such as GDP, inflation, and employment?
2. How does the predictive power of ESG performance compare to that of a traditional market index in forecasting economic growth? (OMX Sustainability Finland GI vs OMX Helsinki GI)
3. What macroeconomic outlook for Finland can be forecast over the next five years using a model incorporating ESG financial performance?

Investigative questions	Theoretical framework (chapter)	Results (chapter)
1. How does Finland's ESG market index correlate with key macroeconomic indicators such as GDP, inflation, and employment?	2.1, 2.2, 2.3, 2.4	3.4.1, 4.1.1
2. How does the predictive power of ESG performance compare to that of a traditional market index in forecasting economic growth? (OMX Sustainability Finland GI vs OMX Helsinki GI)	2.1, 2.2, 2.3, 2.4	3.4.1, 3.4.2 4.1.2
3. What macroeconomic outlook for Finland can be forecast over the next five years using a model incorporating ESG financial performance?	2.1, 2.2, 2.3, 2.4	3.4.3, 4.1.3

Figure 1. Overlay Matrix

1.3 Limitations and Scope

This study focuses on the relationship between ESG financial performance and Finland's macroeconomic trajectory. It is geographically anchored in Finland, using the country as a case study due to its prolonged economic stagnation and the relevance of its small, open economy in the Nordic context. While the core empirical analysis emphasizes national-level economic indicators and domestic ESG index performance, one international ESG benchmark is included as a control variable. This allows the study to better isolate country-specific effects by accounting for global ESG market trends and investor sentiment, without shifting the study's focus beyond Finland.

The temporal scope spans from 2011Q1 to 2024Q4, a period that captures Finland's post-financial-crisis stagnation and the global rise of ESG investing, whilst also being the first full mutual quarter between the chosen variables. Forecasts are limited to medium-term (the next five years), given both the exploratory nature of the analysis and practical data limitations. Moreover, the analysis does not explicitly account for structural breaks caused by major events such as the COVID-19 pandemic or Russia's invasion of Ukraine. Such disruptions may constitute structural breaks in the time series, which could undermine the validity of assumptions regarding the stability of long-run relationships during those intervals.

This research relies on macro-level data, including aggregate economic indicators such as GDP, inflation, and unemployment, alongside financial index performance. It does not incorporate firm-level financials or household data. Similarly, ESG metrics are treated as composite indicators, meaning that their environmental, social, and governance pillars are not analyzed separately. This approach ensures consistency with the study's emphasis on national economic forecasting.

This study intentionally excludes several areas from its scope. The study does not attempt to prove a causal link between ESG performance and economic growth but instead evaluates potential correlations and predictive relationships. It also does not analyze sectoral ESG outcomes, specific government policies, or the microeconomic foundations of growth. These choices reflect a clear focus on macro-level trends, though they also suggest pathways for further research.

1.4 Key Concepts

Employment Rate: The employment rate is a key economic metric that indicates the share of the working-age population currently employed. It serves as a measure of an economy's capacity to provide adequate job opportunities and is determined by dividing the number of employed individuals by the total working-age population (The European Foundation for the Improvement of Living and Working Conditions 2020). A high employment rate suggests robust labor market demand and overall economic strength, whereas a declining rate may point to economic fragility or inadequate job creation. In Finland, Statistics Finland reports the employment rate for individuals aged 20 to 64 (Statistics Finland 2025).

ESG Metrics: Environmental, Social, and Governance (ESG) metrics are indicators used to assess a company's performance across a range of sustainability-related areas. These metrics underpin ESG ratings and scores, which aim to capture how well a company manages its exposure to environmental, social, and governance risks and opportunities. ESG metrics can be input-based, focusing on company policies and practices, or output-based, which measure the actual results of those efforts. While they are increasingly used by investors, policymakers, and stakeholders to evaluate sustainability performance, the coverage and quality of ESG metrics vary significantly across topics and rating providers. (Bijelic, Michel, Mann & Jørgensen 2025, 7-9.)

Financial Indices: Financial indices are statistical measures that use a standardized technique to track the performance of a group of assets, including stocks, bonds, and other instruments. It represents trends in price movements or economic conditions and acts as a benchmark for assessing market segments or investment strategies. (Chen 2024.) Financial indices can be broad-based, like the OMX Helsinki Gross Index (OMXHGI) representing the performance of all shares listed on the Helsinki Stock Exchange, including dividends, or more specialized, such as OMX Sustainability Finland GI (OMXSUSTAINAFIGI), which focuses on Finnish companies that meet specific sustainability criteria and allows investors to assess the performance of firms screened by ESG score (Nasdaq 2025). Although investors are unable to participate directly in an index, they can do so through exchange-traded funds (ETFs) or index funds that mimic its composition. For market analysis, portfolio benchmarking, and passive investing methods, these indices are essential resources. (Chen 2024.)

Gross Domestic Product (GDP): Gross Domestic Product (GDP) represents the total monetary value of all final goods and services produced within a nation's borders during a specific timeframe, typically a quarter or a year. It captures the level of economic activity and indicates the overall scale of a country's economy, making it a key measure of economic performance. Real GDP, which adjusts for inflation or deflation, gives a clearer view of actual growth by isolating changes in output from fluctuations in price levels. (Callen 2019.) This thesis relies on real GDP as the main indicator for assessing Finland's macroeconomic trends, as it provides a stable and accurate measure of economic growth unaffected by price changes.

Inflation: Inflation refers to the rate at which the general prices of goods and services increase over time, leading to a decline in the purchasing power of money. It indicates how much more costly a typical selection of goods and services becomes, usually assessed on an annual basis. When price growth outpaces income growth, individuals can buy less with the same amount of money, effectively decreasing their real standard of living. (Höflmayr 2022, 2.) A common method for tracking inflation is the Consumer Price Index (CPI), which measures changes in the cost of a standardized basket of consumer items. The CPI's percentage change over time reveals how much the cost of living has risen. (Höflmayr 2022, 4-5.)

The Nordic Welfare Model: The Nordic welfare model, as practiced in the Nordic countries including Finland, combines an open, market-based economy with a comprehensive, tax-financed welfare state. It offers universal access to high-quality public services, such as child care, education, health care and elder care, and supports labour-market flexibility through active employment policies and coordinated wage-bargaining. With some of the world's lowest levels of income inequality and eminent social trust, it rests on redistributive taxation and well-functioning democratic institutions. Although it now faces serious pressures from rapid globalization and a sharply ageing population, the model remains widely admired for its successful blend of equity and economic efficiency. (Andersen, Holmström, Honkapohja, Korkman, Söderström & Vartianinen 2007, 11-30.)

Welfare State: A welfare state is a system in which the government provides essential social services such as healthcare, education, and financial support to citizens in need. These services are funded primarily through taxation to promote social well-being and reduce inequality. (Cambridge University Press 2025.)

2 Theoretical Framework

This chapter outlines the theoretical foundations that guide the study's exploration of ESG financial performance and its potential connection to Finland's macroeconomic development. It first introduces Stakeholder Theory, which highlights the importance of managing stakeholder relationships for long-term success. It then discusses Creating Shared Value, which links business competitiveness with societal advancement. The chapter continues by examining the relationship between ESG Performance and Macroeconomic Stability, drawing on recent empirical research. Finally, it considers The Finnish Context, illustrating how Finland's welfare state model may reinforce the connection between ESG metrics and economic growth.

2.1 Stakeholder Theory

Stakeholder Theory, originally formulated by Freeman in 1984 and later developed further by Freeman and McVea (2001), emphasizes that a company's long-term success hinges on how well it manages its relationships with all individuals and groups connected to its objectives (Freeman & McVea 2001, 1-2). In his foundational work, Freeman defined a stakeholder as "any group or individual who is affected by or can affect the achievement of an organization's objectives" (Freeman & McVea 2001, 2). This broad category includes not only shareholders but also employees, customers, suppliers, communities, and government entities (Freeman & McVea 2001, 3). The theory departs from traditional shareholder-focused strategy by promoting a more comprehensive and inclusive perspective. According to Freeman and McVea, organizations that recognize and balance the needs of various stakeholders are better equipped to handle uncertainty, foster legitimacy and trust, and generate enduring value in ways that align ethical responsibility with financial success (Freeman & McVea 2001, 2-4, 9-10).

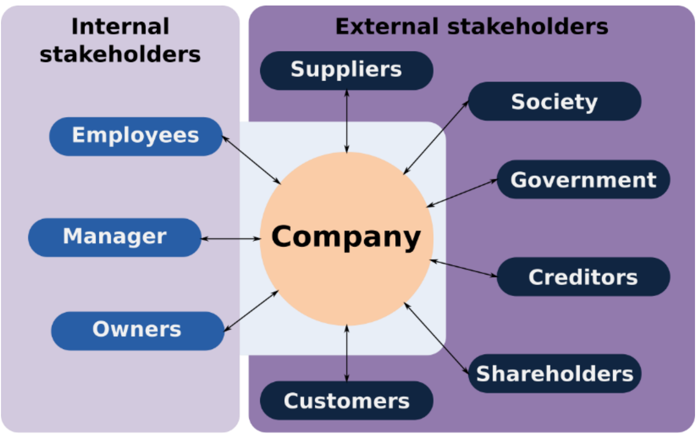


Figure 1. Stakeholder theory. (Grochim 2008)

Under Stakeholder Theory, ESG metrics serve as a practical gauge of how well a company manages its stakeholder relationships in areas of environmental impact, social responsibility, and governance practices (Wang 2024, 1). Strong ESG performance signals that a company is addressing stakeholder concerns in those domains (Wang 2024, 4). According to stakeholder scholars, this can reduce business risks and costs and enhance operational efficiency and innovation (Wang 2024, 3; Jones, Harrison & Felps 2018, 371-391). In other words, good ESG performance is seen as evidence of good stakeholder management, which should translate into more stable and robust financial outcomes for the firm. When scaled to the macroeconomic level, it is reasonable to extrapolate that if a critical mass of firms in an economy (such as Finland's) is highly stakeholder-oriented and ESG-conscious, the result could be fewer systemic shocks (like industrial accidents or social unrest) and a more sustainable macroeconomic growth path. Prior research supports this logic: companies demonstrating strong ESG practices tend to gain buy-in from both external and internal stakeholders, leading to improved brand image, loyalty, and even profitability (Wang 2024, 2-4). This alignment of interests suggests that stakeholder value creation at the firm level can contribute to macro-level stability, providing a theoretical rationale for linking ESG indicators to economic forecasts.

2.2 Creating Shared Value

Creating Shared Value (CSV), articulated by Porter and Kramer in 2011, reinforces the connection between corporate practices and broader economic outcomes. They define shared value as "policies and operating practices that enhance the competitiveness of a company while simultaneously advancing the economic and social conditions in the communities in which it operates" (Porter & Kramer 2011, 6). This concept redefines the purpose of the corporation: instead of viewing social responsibility as a peripheral activity or cost, shared value integrates social and environmental needs into the core business strategy, generating mutual benefits for both the firm and society (Porter & Kramer 2011, 4). At its core, the shared value framework proposes that companies can find business opportunities in addressing societal challenges, such as the development of clean energy technologies, the improvement of workforce skills through education, or the strengthening of local supplier networks, which, in turn, drives innovation and growth (Porter & Kramer 2011, 4-11). Importantly, this approach challenges the conventional view that social progress and economic performance are in conflict, asserting instead that "addressing societal harms and constraints does not necessarily raise costs for firms" and may, in fact, increase productivity and open up new markets (Porter & Kramer 2011, 4-5).

Figure 2. Creating Shared Value. (Shared Value Initiative Hong Kong 2024)

This perspective carries important consequences for how we understand ESG performance as an indicator of economic strength. High ESG ratings typically signal that a company is proactively addressing critical issues such as climate change, resource efficiency, employee welfare, and community engagement; factors closely tied to cost reduction, risk mitigation, and competitive edge (Porter & Kramer 2011, 8-9, 11). For instance, a retail business that cuts waste and energy use not only enhances its environmental profile but also lowers expenses and boosts efficiency (Porter & Kramer 2011, 9). Likewise, firms that prioritize employee development and well-being may see gains in productivity and innovation, while also contributing to the economic stability of their communities (Porter & Kramer 2011, 11). According to Porter and Kramer (2011), the broad implementation of shared value strategies could fuel a new era of innovation and productivity by aligning business success with societal advancement (Porter & Kramer 2011, 4). In the Finnish context, if local companies are widely adopting these practices, robust ESG performance may be an early sign of sustainable economic development. Therefore, CSV offers a compelling framework for interpreting ESG not just as a matter of ethics or reputation, but as a driver of concrete economic benefits that can shape a country's future economic landscape.

2.3 ESG Performance and Macroeconomic Stability

Building on Stakeholder Theory and CSV, a growing body of literature highlights the strong association between ESG performance and long-term economic outcomes. Empirical studies across countries and sectors suggest that sustainable business practices are linked to improved financial performance and reduced downside risk. These outcomes, in turn, support broader macroeconomic stability by enhancing investor confidence, increasing global competitiveness, and mitigating systemic risks. While causal links are still being explored, current evidence underscores ESG's strategic value as a driver of sustainable economic growth. (Koller, Nuttall & Henisz 2019 2-4; Whelan, Ulrich, Van Holt & Clark 2021, 10.) Additionally, corporate investment in environmentally sustainable practices, particularly in high-impact industries, can reduce environmental costs and improve resource utilization efficiency, which in turn enhances financial performance and contributes to enduring economic resilience. (Li, Tang, & Li 2024, 10). Moreover, robust social and governance frameworks, such as commitments to employee well-being, transparency, and anti-corruption, can reduce social inequalities and institutional inefficiencies that typically constrain growth (Diaye, Ho & Oueghlissi 2022, 19-21).

Complementing evidence on how ESG drives broader economic stability, recent studies show that ESG-focused investments themselves are closely intertwined with key macroeconomic indicators. In developed markets, ESG equity securities frequently move in step with sustained economic activity, demonstrating positive coherence with both overall equity-market performance and core sustainability metrics. Although these investments tend to mirror short-term market swings rather than predict them, their consistent alignment with major financial signals makes them a powerful lens through which to evaluate long-term economic health. (Qureshi, Qureshi, Ismail & Yarovaya 2025, 33).

In alignment with stakeholder-focused principles, economies where businesses actively invest in their local communities and uphold the rights of various stakeholders tend to experience greater trust and social cohesion. These qualities contribute to more stable investment environments and support long-term sustainable development. Growing empirical research supports these theoretical connections. For instance, a panel cointegration study covering 29 OECD countries from 1996 to 2014 revealed that although ESG performance had limited short-term impact, it significantly boosted GDP per capita over the long term (Diaye et al. 2022, 19-21). Likewise, in BRICS+ nations (Brazil, Russia, India, China, South Africa, Egypt, Iran, Saudi Arabia, and the UAE) ESG performance has been found to positively influence long-term economic growth, with the original member countries displaying stronger resilience and adaptability to ESG-related disruptions (Algarhi & Karimazondo 2023, 1484-1485).

At the firm level, aggregated evidence from over 2,200 empirical studies reveals a strong business case for ESG: approximately 90 % of studies found a non-negative relationship between ESG and corporate financial performance (Friede, Busch & Bassen 2015, 210, 226-227). The majority of these studies identified a positive relationship, indicating that ESG-aligned companies often outperform or match their peers financially, refuting the notion that sustainability compromises profitability. Additional meta-reviews of post-2015 research further support this view, showing that ESG strategies enhance operational efficiency and offer downside protection during periods of crisis (Whelan et al. 2021, 10).

Together, these findings suggest that ESG performance is not merely a reflection of ethical responsibility, but also a key indicator of economic and financial resilience over the long term.

2.4 The Finnish Context

Finland’s Nordic welfare state model provides a unique institutional context that may amplify the relationship between ESG financial performance and macroeconomic outcomes. The Finnish welfare state is characterized by universal social protections, high public investment in education and healthcare, and a strong emphasis on equality and transparency (Andersen et al. 2007). These features align closely with the principles underlying ESG frameworks, particularly in the areas of social responsibility and governance.

Empirical studies suggest that in Nordic countries, including Finland, there is a positive correlation between ESG performance and financial metrics such as return on equity and net profit margins (Saha & Khan 2024, 809-810). This indicates that firms operating within the Nordic welfare framework may be better positioned to integrate ESG considerations into their business models, leading to enhanced financial performance.

Furthermore, the Finnish government's commitment to sustainability and social welfare has led to the development of policies that support ESG objectives. For instance, Finland's strategic focus on sustainable development and its integration into national policy frameworks reinforce the importance of ESG factors in both public and private sector decision-making. (Finnish National Commission on Sustainable Development 2022.)

In summary, Finland's Nordic welfare state model, with its emphasis on social equity, transparency, and sustainable development, creates a conducive environment for ESG considerations to influence economic outcomes. This institutional context may strengthen the feedback loop between ESG performance and macroeconomic indicators, making ESG metrics a valuable tool for forecasting economic growth in Finland.

3 Empirical Section

This chapter presents the empirical component of the study, detailing the research process from design to results. It begins by clarifying the Research Target and Problem Definition, outlining the core objectives and focus areas. Next, it explains the Method Selection and Justification, highlighting why the specific methodology was chosen for the analysis. The chapter then describes the Work Process and Implementation, covering data collection strategies, data descriptions, and the analytical procedures applied. Finally, it presents the Results and Findings, followed by an Empirical Conclusion that summarizes the key insights drawn from the analysis and discusses their implications for the study's broader objectives.

3.1 Research Target and Problem Definition

This empirical research investigates whether ESG (Environmental, Social, and Governance) financial performance can provide meaningful insights into Finland's economic trajectory. Given the country's persistent low growth and structural challenges since the global financial crisis (Kokkinen et al. 2025), the study aims to test whether sustainable financial indices carry predictive power for key macroeconomic indicators such as GDP, inflation, and employment.

The analysis utilizes three financial indices to structure the empirical investigation. The first is the OMX Sustainability Finland Gross Index (OMXSUSTAINFIGI), which represents Finnish companies that meet defined environmental, social, and governance criteria. This index is free-float adjusted and liquidity screened to ensure investability, providing a sustainability-focused perspective on Finland's equity market. (Nasdaq 2025) The second is the OMX Helsinki Gross Index (OMXHGI), a broad-based benchmark that includes all shares listed on the Helsinki Stock Exchange, reflecting the general performance of the Finnish stock market and serving as a conventional reference point for financial and macroeconomic trends (Nasdaq 2025). The third index incorporated is the MSCI ACWI ESG Leaders Select Sustainability Index, which is based on the MSCI ACWI Index, its parent index, and includes large and mid-cap securities across 23 Developed Markets and 24 Emerging Markets countries. The index aims to represent the performance of an investment strategy that selects securities of companies that have high Environmental, Social, and Governance (ESG) ratings relative to their sector peers and reweights the constituents to meet additional climate requirements imposed (MSCI 2025). All three indices are gross indices, meaning that they not only reflect changes in stock prices, but also any dividends paid by the constituents, making them more robust for econometric analysis (Nasdaq Baltic 2019).

The primary objective is to evaluate the statistical and forecasting relationship between ESG-oriented financial market performance and national economic outcomes, using Finland as a case study. In doing so, the research explores whether ESG metrics can supplement traditional forecasting tools, offering an alternative perspective on long-term economic developments amid structural stagnation and the rising prominence of sustainable finance.

Through comparative econometric modeling, the study examines whether ESG indices provide distinct predictive insights or whether their information content overlaps with that of conventional financial benchmarks, such as the OMX Helsinki GI index. To extend the analysis beyond historical relationships, five-year forecasts (covering 20 quarters) are produced based on the ESG-oriented models. These forecasts aim to offer a forward-looking view of Finland's macroeconomic outlook by leveraging the potential predictive value of ESG financial performance.

3.2 Method Selection and Justification

This study applies the Vector Error Correction Model (VECM) to examine the relationship between ESG financial performance and Finland's macroeconomic indicators, including GDP, inflation, and employment rate. The VECM framework is particularly suited to time series data where variables are non-stationary, meaning that their statistical properties, such as mean and variance, change over time, often due to underlying trends or structural shifts in the economy (Parker 2013, 94; Iordanova 2022). However, if a group of such variables moves together in a consistent way over the long run, they are said to be cointegrated, indicating the presence of a stable equilibrium relationship despite short-term fluctuations (Parker 2013, 94-95).

The VECM is an extension of the Vector Autoregression (VAR) model that explicitly incorporates cointegration structures, allowing it to estimate both short-term adjustments and long-run equilibrium dynamics within a system of interrelated variables (Parker 2013, 95-96). By integrating an error correction mechanism, VECM accounts for how deviations from long-run relationships are corrected over time, enhancing its usefulness in systems where variables are individually non-stationary but move together in the long run.

This makes VECM particularly valuable in macroeconomic analysis, where many economic indicators exhibit non-stationary behavior yet are influenced by common long-term forces. The model's ability to capture both the short-run dynamics and long-run equilibrium relationships also aligns with the forward-looking objectives of this study, which focuses on the predictive role of ESG indices in economic forecasting. Unlike standalone cointegration tests, which identify long-term relationships without modeling short-term dynamics, or Granger causality tests (1969), which capture predictive direction but not equilibrium structure, VECM provides a unified framework that addresses both aspects (Hanck, Arnold, Gerber, & Schmelzer 2024, 394-396; Shojaie & Fox 2021, 291-292). Its capacity to generate multi-period forecasts further supports its selection for this analysis.

The VECM produces three main types of results. The model first reports short-run coefficients on the lagged endogenous variables, showing how past changes in variables such as GDP, inflation, employment, and ESG indices influence current values. For each coefficient, the output includes a standard error (the estimated variability of the coefficient) (The Pennsylvania State University Department of Statistics 2019a), a z-statistic (the coefficient divided by its standard error) (The Pennsylvania State University Department of Statistics 2019b), and a p-value (the probability of observing such a result if the true coefficient were zero), with p-values below common thresholds (e.g., 0.05) indicating statistical significance (Beers 2024). The VECM also provides the cointegration vector (beta), which defines the stable long-run relationship among the variables by showing how deviations from equilibrium are constructed based on the levels of each variable. In addition, the model estimates error correction term (ECT) loadings (alpha coefficients), which measure how quickly variables adjust back toward the long-run equilibrium after a shock, with significant loadings suggesting an active role in restoring balance when the system is out of equilibrium. (Parker 2013, 95-99.)

Before implementing the VECM, it is essential to verify that all variables are integrated of the same order, specifically I(1), which means they become stationary after taking the first difference. To evaluate stationarity, the Augmented Dickey-Fuller (ADF) test, introduced in 1979, was utilized. This test is commonly applied to detect unit roots in time series data, with the null hypothesis suggesting that the data is non-stationary. A sufficiently negative test statistic allows for rejection of the null hypothesis, indicating that the series is indeed stationary. (Cheung & Lai, 1995, 277-280.)

Once stationarity properties are confirmed, the Johansen cointegration test (1988) was used to detect the presence of cointegrating relationships among the variables. As described by Bilgili (1998), this method is preferred over simpler approaches such as the Engle-Granger two-step method because it can identify multiple cointegrating relationships simultaneously and treats all variables as endogenous, meaning that they are determined by their relationship with other variables within the model. This structure is particularly well-suited for economic and financial systems, where feedback loops between variables are common. Importantly, the Johansen test also determines the cointegration rank, which specifies the number of stable long-run relationships present among the variables. (Bilgili 1998, 10.) Determining the correct cointegration rank is crucial, as it defines the appropriate structure of the VECM, balancing between full differencing, which discards long-run information, and modeling in levels, which risks spurious regressions if incorrectly specified (Parker 2013, 98-99).

A critical step in applying the Johansen cointegration test and VECM is determining the correct number of lags to include. Lags are past values of the variables that help capture delayed effects and dynamic relationships over time (Hanck et al. 2024, 313). Choosing the right number of lags is important because too few can miss key dynamics while too many can cause overfitting and loss of efficiency.

To find the best lag length, criteria such as the Akaike Information Criterion (AIC) (Hirotugu Akaike 1973) and the Bayesian Information Criterion (BIC) (Schwarz 1978) were used. Both criteria balance model fit and complexity by adding a penalty for each extra parameter. AIC tends to favor slightly more complex models while BIC favors simpler ones. In practice, several VAR models with different lags are estimated and the lag with the lowest AIC or BIC is selected. (Cavanaugh & Neath 2018, 1-6.)

Finally, in order to evaluate and compare the forecast accuracy of different models, two standard error measures were used: root-mean-square error (RMSE) and mean absolute error (MAE). RMSE is calculated as the square root of the average of the squared differences between predicted and observed values, while MAE is the average of the absolute differences (Hodson, 2022, 5482). Both metrics are widely applied in statistical modeling and forecasting, and each has particular strengths. RMSE has the advantage of penalizing larger errors more heavily because it squares them before averaging, making it especially sensitive to large deviations. It also expresses the error in the same units as the data, which makes interpretation intuitive (Hodson, 2022, 5482-5483). MAE, in contrast, treats all errors equally and is therefore considered more "robust," meaning it is less influenced by extreme values or outliers (Hodson, 2022, 5484).

Choosing between RMSE and MAE depends on the nature of the error distribution: RMSE is optimal when errors are normally distributed, while MAE is preferable when errors tend to have a Laplacian, or double-exponential, distribution with heavier tails (Hodson, 2022, 5483-5484). Because real-world data often contain both normally distributed and non-normal errors, using both RMSE and MAE together provides a more complete and balanced assessment of model performance, as recommended by Chai and Draxler (2014) (Hodson, 2022, 5485).

3.3 Work Process, Data Collection, and Analytical Procedures

The empirical analysis relies on macroeconomic and financial data covering the period from 2011Q1 to 2024Q4. The data preparation process began with ensuring readiness for analysis in Excel. All variables were first visually inspected for consistency, structural patterns, and missing values. Variable values were then averaged to quarterly values to match highest common frequency, and three main datasets were constructed for the analysis. The first includes the Finnish ESG index (OMX Sustainability Finland GI) (Nasdaq 2025). The second combines the Finnish ESG index with a global ESG benchmark, the MSCI ACWI ESG Leaders Index, which serves as a control variable to help isolate Finland-specific effects (MSCI 2025). The third dataset is based on the traditional Finnish stock market index (OMX Helsinki Gross Index, OMXHGI) (Nasdaq 2025), allowing for a comparison between ESG-oriented and conventional financial instruments. The key macroeconomic variables included real GDP (2015 as the reference year), as it isolates the true growth in economic output, unaffected by changes in purchasing power, consumer price index to represent inflation (CPI), and the employment rate among Finnish citizens aged 20-64 (Statistics Finland 2025). All domestic financial index data were transformed into real terms using Finland's national CPI and all indices rebased to 2011=100 to ensure comparability across time. However, it should be noted that the global ESG index remained in nominal terms, as it is not tied to a single country's price level and could not be reliably deflated.

To formally assess the time series properties, ADF test was conducted. Initially, ADF tests were performed at levels, confirming that all variables were non-stationary. Subsequently, ADF tests were applied to the first differences. For most variables, stationarity was achieved at first difference using automatically selected lag lengths based on AIC. For Inflation, additional lag sensitivity analysis was conducted, and stationarity was confirmed at first difference using a manually selected lag based on best test performance. These results validated the assumption that all series are integrated of order one (I(1)) at 95 % confidence level, justifying the application of the VECM framework

Following the stationarity tests, the Johansen cointegration test was employed to determine whether the non-stationary variables were cointegrated, indicating a stable long-run equilibrium relationship despite short-term fluctuations. In this study, the lag length was primarily selected based on the Bayesian Information Criterion (BIC), which recommended a lag length of one for two out of the three models. Although BIC suggested a lag length of eight for the ESG + Control model, the second-lowest BIC was found at lag one, and practical considerations strongly favored the simpler specification. BIC was preferred over AIC because it imposes a heavier penalty on model complexity, making it more appropriate for studies with limited sample sizes, such as this one, which included only 56 quarterly observations per variable. In contrast, AIC continued to decrease with higher lag orders and would have suggested using eight lags for all models, but this would have required estimating an impractically large number of parameters relative to the available data, substantially increasing the risk of overfitting and reducing the reliability of forecasts.

Empirical testing confirmed that using more than one lag quickly led to model deterioration. Models estimated with two lags already showed poorer forecast performance and signs of overfitting, such as increased volatility and unstable long-term dynamics. Models with three lags performed even worse, exhibiting highly erratic behavior and unreliable forecasts. Using eight lags, as suggested by AIC, would have been entirely unrealistic and would have resulted in an extremely overfit and unstable model. Therefore, a lag order of one was selected to ensure a balance between capturing short-term dynamics and maintaining model parsimony.

Once the lag structure was established, the Johansen cointegration tests were conducted using the one-lag specification. At the 95 % confidence level, the tests indicated the appropriate number of cointegration relationships (ranks) for each model. These cointegration ranks were then used as inputs for the estimation of the VECM. In practical terms, using one lag meant that each variable's current quarterly change was modeled as depending on its own and the other variables' previous quarter, along with the deviation from the long-run equilibrium (Parker 2013, 95).

After conducting the VECM test, the forecast accuracy was evaluated using RMSE and MAE. The results of the forecast comparison indicated the relative performance of the ESG-only, ESG+Control, and traditional index models across the selected macroeconomic variables. Visualizations were also produced, comparing the actual values against model forecasts for each specification.

Following the short-term forecast evaluation, long-run forecasts for the next 20 quarters were produced using the ESG-oriented models (ESG-only and ESG+Control). Before estimating the long-run forecasts, the data were standardized to improve model stability and comparability across variables. The scaling procedure transformed each variable to have a mean of zero and a standard deviation of one. After fitting the VECM models on the scaled data, forecasts were generated for the 20-quarter horizon. The forecasts were then inversely transformed using the fitted scaler parameters to make the results interpretable in the original units of measurement. This procedure restored the original scale of the variables.

3.4 Results and Findings

The results of the empirical analysis are organized into three main categories. First, the VECM Tests section presents the short-run and long-run relationships identified through the VECM estimations. Second, the Comparative Tests section evaluates the relative predictive performance of the different models based on standard forecast error metrics. Third, the Forecasts section provides five-year projections for Finland's key macroeconomic variables, utilizing the estimated VECM structures.

3.4.1 VECM Tests

The VECM estimations are presented individually for three model specifications: the ESG-Only model, the ESG + Global Control model, and the Traditional OMX Helsinki Gross Index (OMXHGI) model. Each specification reports the short-run dynamics captured by the lagged endogenous variables, the long-run cointegration relationships among the variables, and the error correction term loadings that describe adjustment speeds toward equilibrium.

The ESG-Only model focuses exclusively on the relationship between Finland's domestic ESG index and macroeconomic indicators. The ESG + Control model incorporates a global ESG benchmark to isolate Finland-specific effects. Finally, the OMXHGI model assesses the relationships between a traditional financial index and macroeconomic outcomes, serving as a baseline for comparison against the ESG-focused models.

Dependent Variable	Lagged Variable	Coefficient	Std. Error	z-Statistic	p-Value	Significance
ESG Index	ESG Index (L1)	0.2937	0.137	2.139	0.032	**Yes**
ESG Index	GDP (L1)	-0.0027	0.002	-1.400	0.161	No
ESG Index	Inflation (L1)	-3.8403	1.758	-2.184	0.029	**Yes**
ESG Index	Employment (L1)	-1.0891	1.208	-0.902	0.367	No
GDP	ESG Index (L1)	20.9203	10.870	1.925	0.054	Marginal
GDP	GDP (L1)	-0.4155	0.150	-2.763	0.006	**Yes**
GDP	Inflation (L1)	-14.4210	139.214	-0.104	0.917	No
GDP	Employment (L1)	127.9306	95.620	1.338	0.181	No
Inflation	ESG Index (L1)	-0.0020	0.008	-0.249	0.803	No
Inflation	GDP (L1)	-0.00000959	0.000	-0.087	0.931	No
Inflation	Inflation (L1)	0.7122	0.102	6.961	0.000	**Yes**
Inflation	Employment (L1)	-0.1568	0.070	-2.231	0.026	**Yes**
Employment	ESG Index (L1)	0.0301	0.013	2.401	0.016	**Yes**
Employment	GDP (L1)	-0.0007	0.000	-4.259	0.000	**Yes**
Employment	Inflation (L1)	0.6408	0.161	3.987	0.000	**Yes**
Employment	Employment (L1)	0.3052	0.110	2.765	0.006	**Yes**

Table 1. Short-Run Coefficients (Lagged Variables) for ESG-Only VECM.

The short-run relationships were captured through the lagged endogenous coefficients. The results showed that the ESG Index was significantly influenced by its own past values, with a positive coefficient of 0.2937 ($p = 0.032$). This indicated that past increases in the ESG Index were associated with continued increases in the subsequent period, suggesting a momentum effect in ESG financial performance. Inflation also exerted a significant influence on the ESG Index in the short run, with a negative coefficient of -3.8403 ($p = 0.029$), indicating that higher inflation levels were associated with a decrease in ESG Index performance during the following quarter. In contrast, GDP and employment did not exhibit statistically significant short-run effects on the ESG Index, with p-values of 0.161 and 0.367, respectively.

In the GDP equation, the past value of the ESG Index showed a positive coefficient of 20.9203 and a p-value of 0.054, suggesting a marginally significant short-run relationship. GDP itself displayed a strong degree of self-dependence, with the lagged GDP coefficient being negative and statistically significant at the 1 % level (coefficient = -0.4155, $p = 0.006$), indicating a correction effect following previous changes. Inflation and employment, however, did not significantly influence GDP in the short run within this model.

When examining inflation as the dependent variable, no significant influence of the ESG Index or GDP was observed. Inflation was primarily driven by its own past values, as indicated by a large positive coefficient of 0.7122 ($p < 0.001$), suggesting strong inflationary persistence over time. The employment rate was also negatively associated with inflation in the short run, with a coefficient of -0.1568 ($p = 0.026$), implying that changes in employment were linked to subsequent inflation dynamics.

Regarding the employment equation, the ESG Index exerted a positive and statistically significant influence, with a coefficient of 0.0301 ($p = 0.016$). This result indicated that higher ESG Index values were associated with higher employment rates in the next quarter. Employment was also significantly influenced by its own lagged values (coefficient = 0.3052, $p = 0.006$), GDP (coefficient = -0.0007, $p < 0.001$), and inflation (coefficient = 0.6408, $p < 0.001$), suggesting that employment dynamics were strongly autocorrelated and responsive to macroeconomic conditions in the short run.

Variable	Coefficient (β)	Std. Error	z-Statistic	p-Value	Significance
ESG Index	1.0000	—	—	—	—
GDP	1.27142	0.456	5.950	0.000	Yes
Inflation	257.3635	85.160	3.022	0.003	Yes
Employment	-3591.4718	499.678	-7.188	0.000	Yes

Table 2. Cointegration Vector (β) for ESG-Only VECM.

Turning to the long-run equilibrium relationships, the cointegration vector estimated a stable association among the variables. The ESG Index was normalized to one, and the long-run coefficients for GDP, inflation, and employment were estimated at 2.7142, 257.3635, and -3591.4718, respectively. This indicated that, in the long-run cointegrating relationship, increases in GDP and inflation were associated with higher ESG Index values, while increases in employment were associated with lower ESG Index values when other factors were held constant.

Equation	ECT Coefficient (α)	Std. Error	z-Statistic	p-Value	Significance
ESG Index	-0.0002	0.000	-0.528	0.598	No
GDP	0.0145	0.032	0.459	0.646	No
Inflation	-0.0000181	0.0000233	-0.777	0.437	No
Employment	0.0002457	0.0000366	6.717	0.000	**Yes**

Table 3. Error Correction Term (ECT) Loadings (α) for ESG-Only VECM.

The error correction term (ECT) loadings, or alpha coefficients, revealed the extent to which variables adjusted back toward the long-run equilibrium following short-run deviations. Among the four variables, only the employment equation showed a statistically significant loading coefficient ($\alpha = 0.0002457$, $p < 0.001$). This suggested that when deviations from the long-run relationship occurred, it was primarily employment that adjusted to restore equilibrium. The ESG Index, GDP, and inflation did not exhibit statistically significant error correction behavior in this model, indicating persistent short-run fluctuations around the long-run equilibrium for these variables.

Dependent Variable	Lagged Variable	Coefficient	Std. Error	z-Statistic	p-Value	Significance
ESG Index	ESG Index (L1)	0.2945	0.164	1.790	0.073	Marginal
ESG Index	GDP (L1)	-0.0028	0.002	-1.479	0.139	No
ESG Index	Inflation (L1)	-3.8958	1.843	-2.114	0.034	**Yes**
ESG Index	Employment (L1)	-0.9938	1.169	-0.850	0.395	No
ESG Index	Global ESG (L1)	0.000008	0.126	0.00006	1.000	No
GDP	ESG Index (L1)	1.94233	12.689	0.743	0.458	No
GDP	GDP (L1)	-0.4273	0.144	-2.975	0.003	**Yes**
GDP	Inflation (L1)	30.4349	142.153	0.214	0.830	No
GDP	Employment (L1)	156.8680	90.213	1.739	0.082	Marginal
GDP	Global ESG (L1)	15.5345	9.718	1.598	0.110	No
Inflation	ESG Index (L1)	-0.0004	0.010	-0.046	0.963	No
Inflation	GDP (L1)	-0.00002	0.000	-0.189	0.850	No
Inflation	Inflation (L1)	0.7097	0.107	6.611	0.000	**Yes**

Inflation	Employment (L1)	-0.1492	0.068	-2.190	0.029	**Yes**
Inflation	Global ESG (L1)	-0.0022	0.007	-0.305	0.760	No
Employment	ESG Index (L1)	0.0160	0.015	1.055	0.291	No
Employment	GDP (L1)	-0.0007	0.000	-3.848	0.000	**Yes**
Employment	Inflation (L1)	0.7653	0.170	4.509	0.000	**Yes**
Employment	Employment (L1)	0.2422	0.108	2.249	0.025	**Yes**
Employment	Global ESG (L1)	0.0176	0.012	1.515	0.130	No
Global ESG	ESG Index (L1)	0.2295	0.203	1.131	0.258	No
Global ESG	GDP (L1)	0.0014	0.002	0.628	0.530	No
Global ESG	Inflation (L1)	-4.5642	2.273	-2.008	0.045	**Yes**
Global ESG	Employment (L1)	-1.9998	1.442	-1.387	0.166	No
Global ESG	Global ESG (L1)	0.0978	0.155	0.629	0.529	No

Table 4. Short-Run Coefficients (Lagged Variables) for ESG + Control VECM.

The short-run dynamics, reflected in the lagged coefficients, showed that the Finnish ESG Index continued to exhibit positive dependence on its own past values, with a coefficient of 0.2945 and a p-value of 0.073. Although the significance level had weakened slightly compared to the ESG-only model, it remained marginally notable. The inflation variable maintained a statistically significant negative effect on the ESG Index in the short run, with a coefficient of -3.8958 and a p-value of 0.034. This was consistent with the earlier model, indicating that inflationary pressures were associated with declines in ESG financial performance in Finland. GDP and employment did not exert statistically significant short-run effects on the ESG Index, and the global ESG control variable itself (Global ESG) showed an almost null coefficient with an extremely high p-value, indicating that global ESG movements did not materially affect short-term changes in the Finnish ESG Index.

In the GDP equation, the influence of the Finnish ESG Index diminished substantially when the global ESG variable was included. The coefficient of the Finnish ESG Index on GDP fell to 9.4233 with a p-value of 0.458, compared to marginal significance in the ESG-only model. This suggested that part of the earlier observed relationship may have been influenced by global ESG factors. GDP's own lagged value remained significantly negative (coefficient = -0.4273, p = 0.003), indicating strong mean-reversion behavior over time. Employment exhibited a positive but only marginally significant relationship with GDP (p = 0.082), while neither inflation nor the global ESG index exerted statistically significant short-run effects.

Inflation continued to display high persistence, with its own lagged coefficient remaining large and significant (coefficient = 0.7097, p < 0.001). No significant short-run relationship between the ESG Index and inflation was observed. The employment rate was again negatively associated with inflation in the short run, with a coefficient of -0.1492 (p = 0.029), suggesting that employment dynamics continued to play a role in short-term inflation behavior.

The employment equation revealed notable changes with the inclusion of the global control. The short-run positive effect of the Finnish ESG Index on employment, which had been statistically significant in the ESG-only model, weakened and became insignificant (coefficient = 0.0160, p = 0.291). This change suggested that the earlier relationship may have partly reflected global ESG market influences rather than being purely domestic. However, GDP, inflation, and employment's own lagged values continued to significantly influence employment dynamics. Specifically, GDP maintained a strong negative effect (coefficient = -0.0007, p < 0.001), inflation showed a positive effect (coefficient = 0.7653, p < 0.001), and employment itself remained significantly autocorrelated (coefficient = 0.2422, p = 0.025).

The new equation introduced for the Global ESG Index showed that its short-run dynamics were not significantly influenced by the Finnish ESG Index, GDP, or employment. However, inflation had a statistically significant negative effect on the Global ESG Index (coefficient = -4.5642, p = 0.045), suggesting that global ESG financial performance was also vulnerable to inflationary pressures.

Variable	Coefficient (β)	Std. Error	z-Statistic	p-Value	Significance
ESG Index	1.0000	—	—	—	—
GDP	0.1481	0.034	4.344	0.000	**Yes**
Inflation	7.1082	8.756	0.812	0.417	No
Employment	-225.2682	31.137	-7.235	0.000	**Yes**
Global ESG	0.0176	0.885	0.020	0.984	No

Table 5. Cointegration Vector (β) for ESG + Control VECM.

The cointegration vector, which captured the long-run equilibrium relationship among the variables, showed that the Finnish ESG Index maintained a stable positive association with GDP and inflation, and a negative association with employment. The estimated coefficients for the cointegration relationship were 0.1481 for GDP, 7.1082 for inflation, and -225.2682 for employment. The coefficient on the global ESG control variable was very small (0.0176) and statistically insignificant in the long-run equilibrium, indicating that, over the long term, the Finnish ESG Index was largely independent of global ESG movements when controlling for domestic macroeconomic fundamentals.

Equation	ECT Coefficient (α)	Std. Error	z-Statistic	p-Value	Significance
ESG Index	-0.0026	0.006	-0.453	0.650	No

GDP	0.5044	0.449	1.124	0.261	No
Inflation	-0.0002	0.000	-0.635	0.525	No
Employment	0.0035	0.001	6.584	0.000	**Yes**
Global ESG	-0.0182	0.007	-2.532	0.011	**Yes**

Table 6. Error Correction Term (ECT) Loadings (α) for ESG + Control VECM.

The error correction term loadings (alpha coefficients) revealed mixed adjustment dynamics. In the employment equation, the adjustment coefficient was positive and highly significant ($\alpha = 0.0035$, $p < 0.001$), suggesting that employment actively corrected toward the long-run equilibrium after deviations. This result was consistent with the ESG-only model. However, the ESG Index itself, GDP, and inflation did not show statistically significant adjustment speeds, indicating persistent deviations from equilibrium over the short term for these variables. Interestingly, the Global ESG Index showed a significant negative adjustment coefficient ($\alpha = -0.0182$, $p = 0.011$), implying that the global ESG benchmark tended to correct back to equilibrium after shocks, whereas the domestic ESG Index did not.

Dependent Variable	Lagged Variable	Coefficient	Std. Error	z-Statistic	p-Value	Significance
OMXHGI	OMXHGI (L1)	0.3870	0.137	2.828	0.005	**Yes**
OMXHGI	GDP (L1)	-0.0033	0.002	-1.663	0.096	Marginal
OMXHGI	Inflation (L1)	-3.7356	1.841	-2.030	0.042	**Yes**
OMXHGI	Employment (L1)	-1.0748	1.254	-0.857	0.391	No
GDP	OMXHGI (L1)	18.4150	10.515	1.751	0.080	Marginal
GDP	GDP (L1)	-0.4176	0.152	-2.747	0.006	**Yes**
GDP	Inflation (L1)	-5.6930	141.412	-0.040	0.968	No
GDP	Employment (L1)	125.5167	96.359	1.303	0.193	No
Inflation	OMXHGI (L1)	-0.0035	0.008	-0.452	0.651	No
Inflation	GDP (L1)	-0.00000106	0.000	-0.010	0.992	No
Inflation	Inflation (L1)	0.7050	0.103	6.823	0.000	**Yes**
Inflation	Employment (L1)	-0.1554	0.070	-2.207	0.027	**Yes**
Employment	OMXHGI (L1)	0.0300	0.012	2.508	0.012	**Yes**
Employment	GDP (L1)	-0.0008	0.000	-4.348	0.000	**Yes**
Employment	Inflation (L1)	0.6628	0.161	4.115	0.000	**Yes**
Employment	Employment (L1)	0.2986	0.110	2.721	0.007	**Yes**

Table 7. Short-Run Coefficients (Lagged Variables) for OMXH VECM.

The short-run dynamics captured by the lagged endogenous coefficients showed that the OMX Helsinki Gross Index was significantly influenced by its own past values. The lagged coefficient on the OMXHGI was 0.3870 with a p-value of 0.005, indicating a strong and statistically significant persistence in the index's performance over short periods. This suggested that the traditional stock market tended to exhibit momentum effects similar to those observed for the ESG-based indices.

Inflation continued to play a significant role in the short-run dynamics of the stock market index, as evidenced by a negative coefficient of -3.7356 ($p = 0.042$) in the OMXHGI equation. This implied that inflationary pressures were associated with declines in the traditional stock index in the short run. GDP and employment did not significantly influence the OMXHGI in the short term, although the lagged GDP variable had a p-value of 0.096, indicating a marginal relationship.

In the GDP equation, the lagged OMXHGI showed a positive coefficient of 18.4150 with a p-value of 0.080. While this fell short of conventional significance thresholds, it suggested a potential short-run link between traditional stock market performance and GDP growth, similar to the patterns observed in the ESG-based models. GDP again demonstrated a strong mean-reversion tendency, with its own lagged value being negative and statistically significant (coefficient = -0.4176, $p = 0.006$). Inflation and employment did not have significant short-run effects on GDP in this model.

The inflation equation maintained strong persistence, with the lagged inflation coefficient at 0.7050 and highly significant ($p < 0.001$). Neither the OMXHGI nor GDP exerted a meaningful short-run influence on inflation, while employment continued to be negatively associated with inflation (coefficient = -0.1554, $p = 0.027$), replicating the pattern found in previous model specifications.

In the employment equation, the traditional stock market index had a positive and statistically significant short-run effect (coefficient = 0.0300, $p = 0.012$). This result indicated that increases in the OMXHGI were associated with rises in employment during the following quarter. GDP and inflation also exerted significant influences on employment, with GDP displaying a negative relationship (coefficient = -0.0008, $p < 0.001$) and inflation a positive one (coefficient = 0.6628, $p < 0.001$). Employment was strongly autocorrelated, with its own lagged value being statistically significant at conventional levels (coefficient = 0.2986, $p = 0.007$).

Variable	Coefficient (β)	Std. Error	z-Statistic	p-Value	Significance
OMXHGI	1.0000	—	—	—	—
GDP	0.7257	0.124	5.854	0.000	**Yes**
Inflation	73.9169	22.996	3.214	0.001	**Yes**
Employment	-993.0440	135.537	-7.327	0.000	**Yes**

Table 8. Cointegration Vector (β) for OMXH VECM.

Turning to the long-run relationships, the cointegration vector estimated a stable equilibrium association between the variables. In this model, the OMX Helsinki Gross Index was normalized to one, and the estimated long-run coefficients were 0.7257 for GDP, 73.9169 for inflation, and -993.0440 for employment. This indicated that over the long term, the traditional Finnish stock market index tended to move positively with GDP and inflation, and negatively with employment, when other factors were held constant.

Equation	ECT Coefficient (α)	Std. Error	z-Statistic	p-Value	Significance
OMXHGI	-0.0007	0.002	-0.440	0.660	No
GDP	0.0637	0.116	0.551	0.582	No
Inflation	-0.0000634	0.0000845	-0.751	0.453	No
Employment	0.0009	0.000	6.783	0.000	**Yes**

Table 9. Error Correction Term (ECT) Loadings (α) for OMXH VECM.

The error correction term loadings (alpha coefficients) provided further insights into the adjustment speeds toward long-run equilibrium. Among the four variables, only the employment equation showed a statistically significant adjustment loading ($\alpha = 0.0009$, $p < 0.001$). This result suggested that employment actively corrected toward the long-run equilibrium when deviations occurred, while the OMXHGI, GDP, and inflation variables did not show statistically significant adjustment speeds, indicating persistent short-term deviations from equilibrium.

3.4.2 Comparative Tests

To assess the predictive performance of the three VECM models, ESG-only, ESG + Global Control, and Traditional (OMX Helsinki Gross Index), out-of-sample forecasts were evaluated using two standard error metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). These metrics summarize the deviation between predicted and actual values, with lower values indicating greater predictive accuracy.

Variable	ESG RMSE	ESG + Control RMSE	Traditional RMSE	ESG MAE	ESG + Control MAE	Traditional MAE
Employment	1.004	1.923	**0.933**	0.809	1.792	**0.724**
GDP	**277.822**	305.814	315.299	247.673	**226.929**	284.095
Inflation	1.696	**1.625**	1.759	1.448	**1.428**	1.508

Table 10. Forecast Error Metrics (RMSE and MAE) by Model and Macroeconomic Variable

Forecast results for employment indicated that the Traditional model produced the lowest prediction error across both RMSE and MAE. The ESG-only model performed moderately, while the ESG + Global Control model showed higher error levels.

For GDP, the ESG-only model yielded the lowest RMSE, whereas the ESG + Control model produced the lowest MAE. The Traditional model showed the highest forecast error by both metrics for GDP.

In terms of inflation, the ESG + Global Control model achieved the lowest forecast errors across both metrics. The ESG-only and Traditional models produced similar levels of inflation forecast accuracy, with the Traditional model yielding slightly higher RMSE and MAE values.

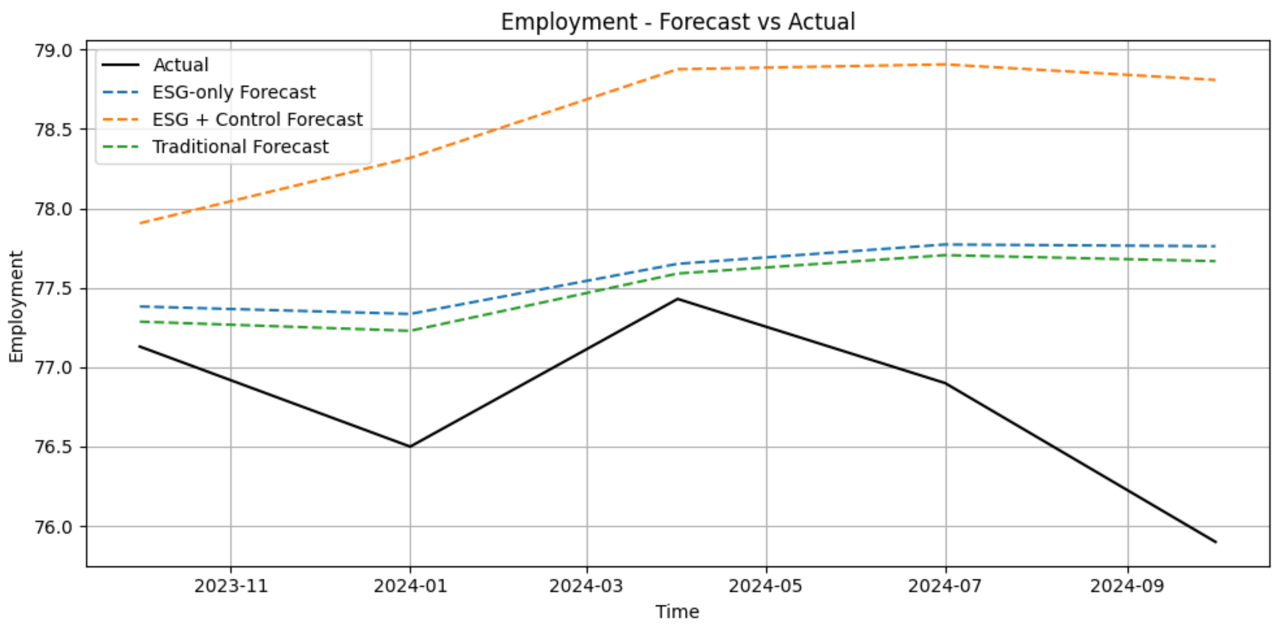


Figure 3. Employment - Forecast vs Actual

Figure 3 compares the actual employment rate with forecasts from the three VECM models. All models overestimated employment relative to observed values. The ESG + Global Control model produced the highest trajectory, diverging most notably from the actual series, particularly in the second half of the forecast period. The ESG-only and Traditional models generated similar paths, both remaining closer to the observed values. Among these, the Traditional model tracked the actual employment data most closely, especially in the later periods.

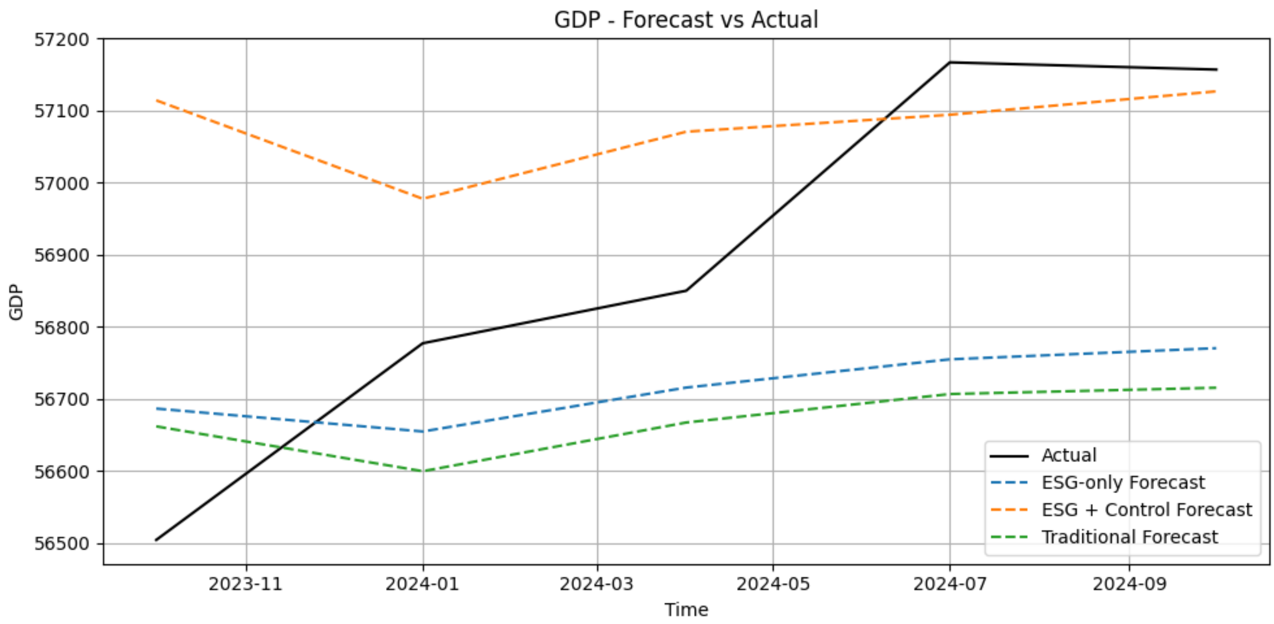


Figure 4: GDP - Forecast vs Actual

Figure 4 shows the real GDP forecasts in comparison to the actual series. While the actual GDP followed a rising trend throughout the forecast window, all models underestimated its level. The ESG + Global Control model remained consistently above the ESG-only and Traditional forecasts but still below the actual data. Both the ESG-only and Traditional models predicted more subdued GDP growth, with the ESG-only model slightly outperforming the Traditional model in the final quarters of the period.

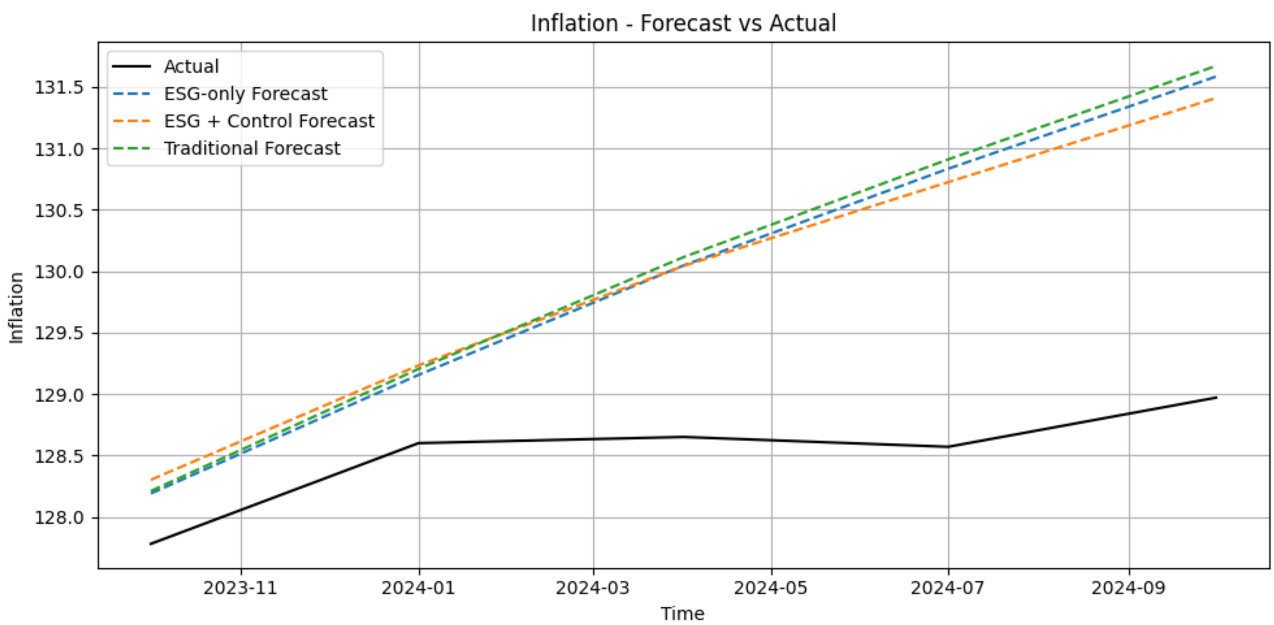


Figure 5: Inflation - Forecast vs Actual, 2011 = 100

Figure 5 presents a comparison between actual inflation values and forecasts from the three models. All forecasts overestimated inflation during the evaluation period. Among them, the ESG + Global Control forecast remained closest to the actual series throughout the forecast horizon, while the Traditional model showed the greatest deviation in the latter periods. The ESG-only model also overestimated inflation, though to a slightly lesser degree than the Traditional forecast.

3.4.3 Forecasts

To extend the analysis beyond in-sample model fit and short-run dynamics, out-of-sample forecasts were generated for a five-year period (2025Q1–2029Q4) using both the ESG-only and ESG + Global Control model specifications. These forecasts provide a forward-looking assessment of the likely macroeconomic path of Finland, conditioned on ESG financial performance and historical macroeconomic behavior. The focus is on three national-level indicators: real GDP, consumer price inflation (CPI), and the employment rate.

Quarter	ESG_Index	GDP	Inflation	Employment
2025Q1	207.16	57142.22	129.49	76.80
2025Q2	207.99	57373.74	129.84	77.66
2025Q3	208.98	57441.48	130.13	77.60
2025Q4	211.61	57471.66	130.51	77.46
2026Q1	214.43	57549.69	130.95	77.60
2026Q2	216.68	57643.62	131.39	77.79
2026Q3	218.71	57721.15	131.84	77.89
2026Q4	220.83	57789.24	132.29	77.96
2027Q1	222.97	57859.57	132.76	78.06
2027Q2	225.02	57931.58	133.24	78.16
2027Q3	227.02	58001.72	133.73	78.25
2027Q4	229.01	58070.02	134.22	78.34
2028Q1	230.98	58137.83	134.71	78.43
2028Q2	232.94	58205.46	135.21	78.52
2028Q3	234.87	58272.64	135.71	78.61
2028Q4	236.80	58339.35	136.21	78.70
2029Q1	238.72	58405.77	136.72	78.79
2029Q2	240.63	58471.99	137.22	78.88
2029Q3	242.54	58538.02	137.73	78.97
2029Q4	244.44	58603.89	138.24	79.05

Table 11. Forecast (ESG-only) - Next 20 Quarters

The ESG-only model, based solely on domestic ESG index dynamics, forecasts a gradually improving macroeconomic environment. Real GDP rises steadily from 57 142 million EUR in 2025Q1 to 58 604 million EUR by the end of 2029. This translates to an average quarterly GDP growth rate of roughly 0.25 %, reflecting a stable, low-growth recovery pattern.

Inflation is projected to climb from 129.49 to 138.24, indicating a moderate but persistent price level increase over the period. The employment rate also trends upward, from 76.80 % to 79.05 %, showing gradual improvement in labor market conditions.

Quarter	ESG_Index	GDP	Inflation	Employment	Global_ESG
2025Q1	207.77	57420.65	129.50	76.46	499.01
2025Q2	208.64	57540.40	129.93	76.78	504.73
2025Q3	210.14	57595.50	130.35	76.81	511.41
2025Q4	212.32	57656.13	130.81	76.85	518.75
2026Q1	214.53	57736.60	131.28	76.96	526.05
2026Q2	216.57	57815.69	131.76	77.06	533.28
2026Q3	218.57	57889.12	132.24	77.14	540.55
2026Q4	220.57	57961.34	132.74	77.22	547.86
2027Q1	222.55	58033.77	133.23	77.30	555.17
2027Q2	224.50	58105.55	133.74	77.38	562.47
2027Q3	226.44	58176.45	134.24	77.46	569.77
2027Q4	228.37	58246.79	134.75	77.53	577.07
2028Q1	230.28	58316.76	135.26	77.61	584.38
2028Q2	232.19	58386.38	135.78	77.68	591.68
2028Q3	234.08	58455.69	136.29	77.75	598.98
2028Q4	235.98	58524.75	136.81	77.83	606.29
2029Q1	237.86	58593.60	137.33	77.90	613.59
2029Q2	239.74	58662.29	137.85	77.97	620.90
2029Q3	241.62	58730.83	138.37	78.04	628.21
2029Q4	243.49	58799.26	138.89	78.11	635.51

Table 12. Forecast (ESG + Control) - Next 20 Quarters

When incorporating the MSCI ACWI ESG Leaders Index as a global control variable, the fore-casted macroeconomic indicators exhibit slightly different trajectories. GDP is projected to rise from 57 421 million EUR to 58 799 million EUR, slightly higher than the ESG-only model by the end of 2029, suggesting that incorporating global ESG sentiment slightly strengthens the growth outlook.

Inflation in this model also increases steadily, from 129.50 to 138.89, a marginally higher endpoint than under the ESG-only scenario. However, the employment forecast is more conservative: the rate rises from 76.46 % to 78.11 %, almost a full percentage point lower than in the ESG-only model, suggesting more modest labor market improvements when global financial dynamics are taken into account.

These quantitative projections are complemented by visual comparisons that place the forecasts in the context of recent macroeconomic developments. Figures 6-8 plot both historical observations (from 2022 onward) and the forecasted values generated by the ESG-only and ESG + Control models. This visualization highlights not only the general direction of projected change but also the relative differences in level and trajectory between the two model specifications.

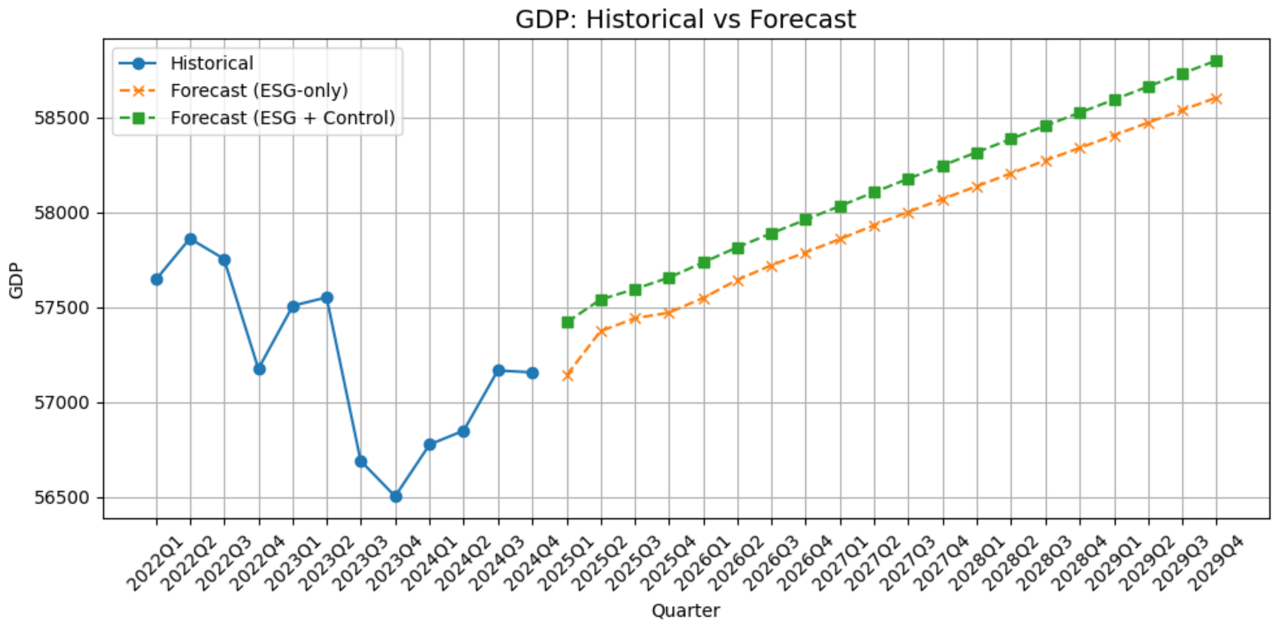


Figure 6: GDP - Historical vs Forecast

Figure 6 illustrates the expected path of GDP until Q4 of 2029. The upward trend in both forecast scenarios contrasts with the volatility and downward trend of the GDP movement observed in recent years. The ESG + Control model consistently predicts marginally higher GDP values than the ESG-only model, with the latter suggesting a more modest recovery.

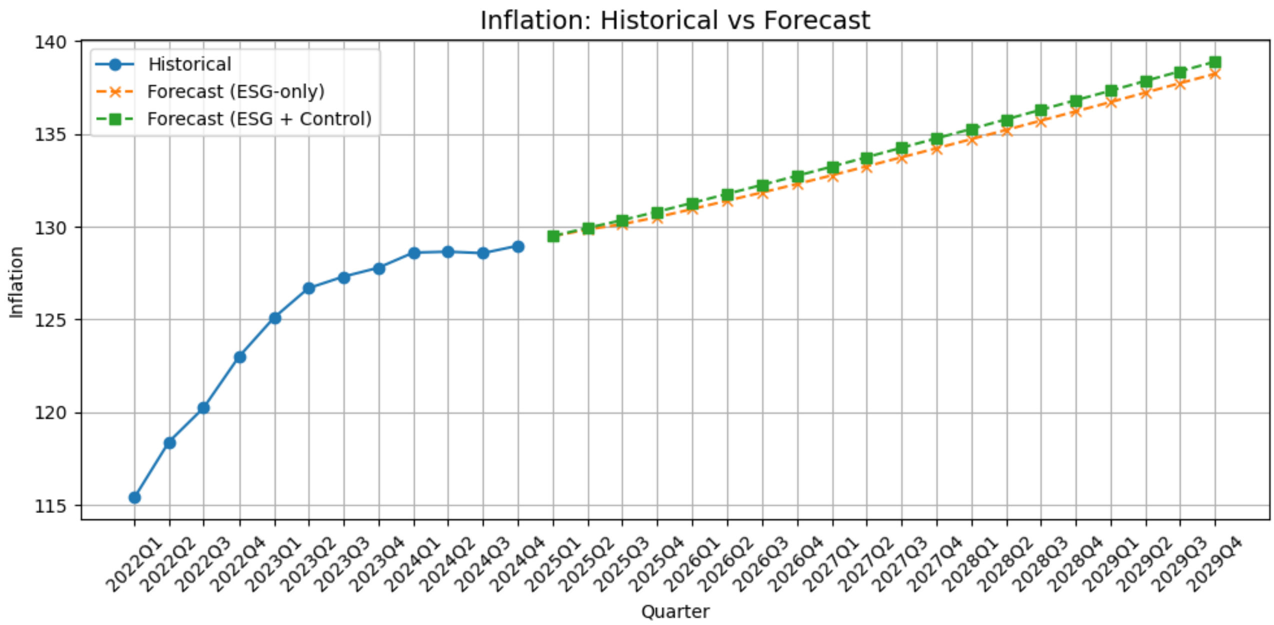


Figure 7: Inflation - Historical vs Forecast, 2011 = 100

A similar pattern emerges in Figure 7, which shows inflation forecasts relative to historical CPI levels. While both models anticipate persistent inflationary pressure, the ESG + Control forecast produces slightly higher inflation estimates throughout the five-year period.

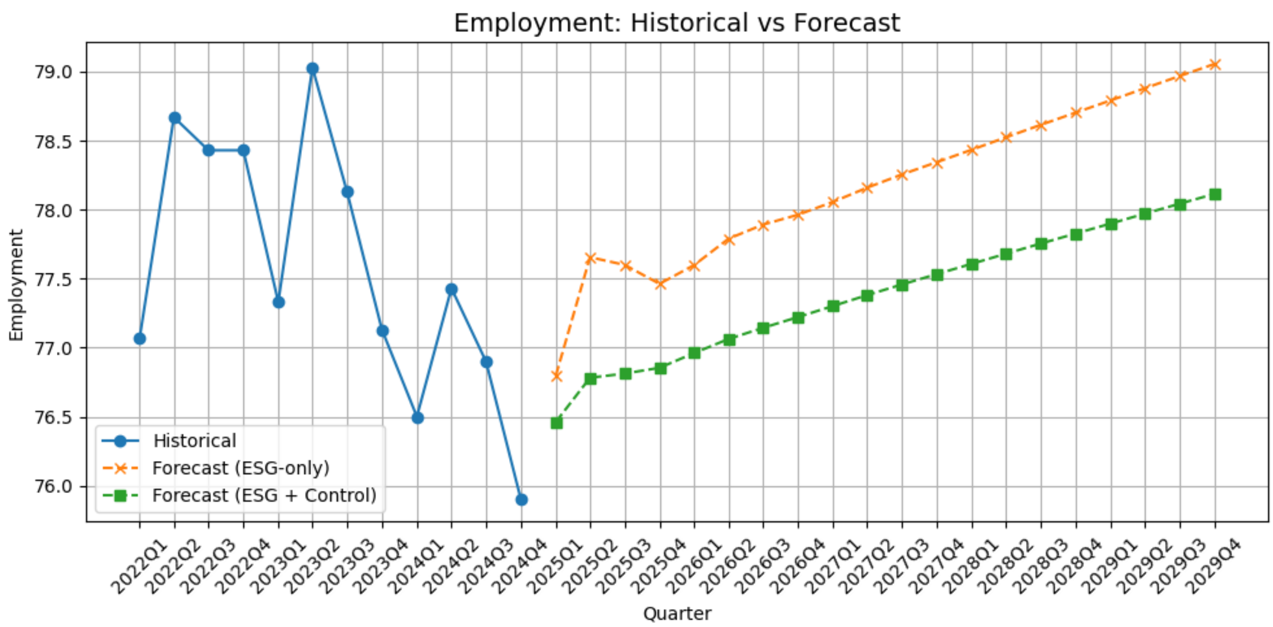


Figure 8: Employment - Historical vs Forecast

Finally, Figure 8 presents the forecasted employment rate. Both forecasts starkly contrast with the recent negative trend in the employment rate. The ESG-only model suggests a more robust recovery, with employment rising to 79.05 % by 2029Q4. The ESG + Control model, while still projecting improvement, levels off closer to 78.11 %, suggesting more modest labor market gains. This divergence underscores the sensitivity of employment projections to the structure of the underlying financial signal used in the forecasting model.

4 Discussion

This chapter provides a comprehensive discussion of the research findings in relation to the central research question of whether ESG financial market performance can be effectively utilized to predict economic growth in Finland, and the three supporting investigative questions. It begins by interpreting the results of the empirical analysis, examining how ESG indices relate to key macroeconomic indicators such as GDP, inflation, and employment. These results are evaluated through the lens of the study's theoretical framework, particularly Stakeholder Theory (Freeman 1984) and Creating Shared Value (Porter & Kramer 2011), to assess whether ESG metrics offer meaningful insight into macroeconomic dynamics. The chapter then critically evaluates the methodological reliability and validity of the study, addressing how well the chosen approach supported the investigation of these questions. This is followed by broader conclusions that synthesize the study's contributions, offer practical implications, and propose areas for further research. Finally, the chapter reflects on the thesis process from a personal perspective, highlighting key learning outcomes, challenges faced, and skills developed throughout the journey.

4.1 Examination and Interpretation of Results

This section critically examines the empirical findings in light of the theoretical framework and in relation to the central research question: Can ESG financial market performance be utilized effectively to predict economic growth in Finland? The discussion also addresses the three supporting investigative questions by evaluating the empirical evidence from the three VECM model specifications: ESG-Only, ESG + Global Control, and the Traditional OMX Helsinki Gross Index (OMXHGI) model. The interpretation follows a logical structure, beginning with direct ESG-macroeconomic relationships, moving to comparative predictive performance, and concluding with forward-looking forecasts.

4.1.1 IQ1. Correlation With Macroeconomic Variables

To address the first investigative question, "How does Finland's ESG market index correlate with key macroeconomic indicators such as GDP, inflation, and employment?", the ESG-Only VECM reveals both promising relationships and important caveats. In the short run, Inflation emerges as a significant negative influence on the ESG Index (coefficient = -3.8403, $p = 0.029$), highlighting ESG investors' sensitivity to price instability. Employment shows a statistically significant positive relationship with the ESG Index in the short run (coefficient = 0.0301, $p = 0.016$), suggesting that rising ESG performance may signal or even support short-term job creation, possibly due to market optimism around firms with strong social and governance practices. GDP shows a marginally significant short-run relationship with the ESG Index (coefficient = 20.9203, $p = 0.054$), but does not reach the conventional 5 % significance threshold. Taken together, these results suggest that Finland's ESG index both reacts to macroeconomic shocks, particularly inflation, and contains some predictive information, especially regarding employment.

These findings align with both Stakeholder Theory and CSV, showing that ESG performance reflects firms' responsiveness to stakeholder welfare and their ability to create economic and social value (Freeman & McVea 2001, 1-10; Porter & Kramer 2011, 4-11). The short-run effects of inflation and employment support this, while the marginal GDP impact is consistent with both theories' long-term focus. Broader studies support ESG's positive links to long-term financial performance, reduced risk, and macroeconomic resilience, whilst also exhibiting partly similar results in terms of respond-anticipate-relationship (Koller et al. 2019 2-4; Whelan et al. 2021, 10; Diaye et al. 2022, 19-21; Qureshi et al. 2025, 33, Algarhi & Karimazondo 2023, 1484-1485).

The long-run cointegration analysis provides a more nuanced understanding of these relationships. When the ESG Index is normalized to one, the long-run coefficients for GDP, inflation, and employment are 1.27142 ($p < 0.001$), 257.3635 ($p = 0.003$), and -3591.4718 ($p < 0.001$), respectively. These results suggest that in equilibrium, ESG performance is positively associated with both GDP and inflation. This is in line with both Stakeholder Theory and CSV, as well as broader studies, through consistent long-term value creation and adaptation to societal challenges (Freeman & McVea 2001, 1-10; Porter & Kramer 2011, 4-11; Koller et al. 2019 2-4; Whelan et al. 2021, 10; Diaye et al. 2022, 19-21; Qureshi et al. 2025, 33, Algarhi & Karimazondo 2023, 1484-1485). However, ESG performance is negatively associated with employment. This juxtaposition, a positive short-run effect of ESG on employment, yet a strongly negative long-run relationship, raises important questions about the structural nature of ESG-aligned firms in Finland. One plausible interpretation is that ESG-leading firms may engage in job creation or sustainability reporting practices that positively influence employment perceptions in the short term, but over the long term tend to be more capital-intensive or technologically advanced, relying on automation and digitalization rather than labor expansion. These firms may prioritize efficiency, innovation, or shareholder value under ESG frameworks that do not necessarily require proportional increases in headcount. Whilst contradicting the core ideas of Stakeholder Theory and CSV in terms of employee value creation, the proposition is still partially supported by Porter & Kramer's claims of increased efficiency under prominent Shared Value circumstances (Porter & Kramer 2011, 11).

The error correction term (ECT) coefficients shed light on how quickly variables return to equilibrium after short-run deviations. Among the variables tested, only employment shows statistically significant adjustment behavior ($\alpha = 0.0002457$, $p < 0.001$), suggesting that it is the primary channel through which long-run equilibrium is restored. This is likely because employment is inherently more flexible and responsive to economic shifts. In contrast, the ESG Index, GDP, and inflation do not exhibit significant adjustment speeds, implying that short-term deviations in these variables may persist without immediate correction.

With the introduction of the global ESG control variable in the second model specification, the dynamics observed in the ESG-Only model shift significantly. The ESG + Global Control model incorporates the MSCI ACWI ESG Leaders Index to account for the influence of international ESG sentiment. In this expanded framework, the domestic ESG Index no longer demonstrates a statistically significant short-run effect on either GDP (coefficient = 1.94233, p = 0.458) or employment (coefficient = 0.0160, p = 0.291), suggesting that the previously identified domestic relationships may have been partly driven by global ESG trends, which aligns with the notion that shared value stems from global interconnectedness (Porter & Kramer 2011, 4-7). Notably, the global ESG index itself shows virtually no explanatory power across Finnish macroeconomic variables, with extremely high p-values and near-zero coefficients, such as in the ESG equation (coefficient = 0.000008, p = 1.000). Despite these shifts, the negative short-run impact of inflation on ESG performance remains statistically significant (coefficient = -3.8958, p = 0.034), reinforcing the conclusion that ESG financial markets in Finland are consistently sensitive to inflationary pressures.

In the long-run cointegration vector of the ESG + Global Control model, the ESG Index remains positively associated with GDP (coefficient = 0.1481, p < 0.001) and negatively with employment (coefficient = -225.2682, p < 0.001), while inflation no longer has a statistically significant effect (p = 0.417). The global ESG index continues to lack long-term influence (coefficient = 0.0176, p = 0.984). The ECT results confirm earlier findings, with only employment adjusting significantly to deviations from the long-run equilibrium (α = 0.0035, p < 0.001).

4.1.2 IQ2. Comparative Results Versus Traditional Index

The second investigative question asks: “How does the predictive power of ESG performance compare to that of a traditional market index in forecasting economic growth? (OMX Sustainability Finland GI vs OMX Helsinki GI)”. The short-run dynamics of the Traditional OMXHGI model closely resemble those of the ESG-Only model. The index is significantly autocorrelated (coefficient = 0.3870, p = 0.005), negatively influenced by inflation (coefficient = -3.7356, p = 0.042), and positively related to employment (coefficient = 0.0300, p = 0.012). GDP shows a marginal effect (coefficient = 18.4150, p = 0.080), similar to the ESG-Only model’s results. These similarities suggest that both ESG and traditional financial indices capture overlapping short-run economic signals.

In the long run, the OMXHGI is positively associated with GDP (coefficient = 0.7257, p < 0.001) and inflation (coefficient = 73.9169, p = 0.001), and negatively associated with employment (coefficient = -993.0440, $p < 0.001$). These results align closely with the ESG cointegration vectors, reinforcing the interpretation that broad financial market indices, whether ESG-filtered or not, tend to reflect the same macroeconomic forces. The ECT analysis confirms that, once again, only employment adjusts significantly toward equilibrium ($\alpha = 0.0009$, $p < 0.001$).

More revealing are the models' forecast performance, assessed using RMSE and MAE metrics. For employment, the Traditional model achieves the lowest forecast errors (RMSE = 0.933; MAE = 0.724), outperforming both ESG-based models. For GDP, the ESG-Only model produces the lowest RMSE (277.822), while the ESG + Global Control model yields the lowest MAE (226.929), slightly outperforming the traditional index. For inflation, the ESG + Global Control model is the most accurate overall (RMSE = 1.625; MAE = 1.428). These results show that ESG-based models offer limited predictive advantage over traditional indices. While they perform slightly better for GDP and inflation forecasting, they fall short in employment prediction.

Overall, these findings reinforce the sentiment that ESG-based models are competitive with traditional financial indices and occasionally outperform them, particularly in forecasting GDP and inflation. This supports empirical research such as Friede et al. (2015, 210, 226-227), which finds that the majority of ESG-aligned companies match or outperform their peers financially, thereby refuting the notion that sustainability compromises profitability.

The similarity in results between ESG and traditional indices likely reflects how deeply ESG principles are already embedded in Finnish society and corporate landscape. In Finland’s institutional context, characterized by transparency, social equity, and sustainability, even firms in traditional indices likely often align with ESG values. This reduces the distinction between ESG and non-ESG investments and may explain the limited added predictive power of ESG-specific indices.

However, this does not undermine the relevance of Stakeholder Theory or CSV. Instead, it reinforces them. According to these frameworks, economies that widely embrace ESG-aligned practices, like Finland, should experience greater macroeconomic stability and sustainable growth. The convergence of ESG and traditional index behavior in Finland is a sign that stakeholder-centric ideology and shared value principles are already deeply rooted into mainstream, making ESG metrics less of a differentiator. (Freeman & McVea 2001, 1-10; Porter & Kramer 2011, 4-11; Finnish National Commission on Sustainable Development 2022.)

4.1.3 IQ3. Five-Year Forecasts

To answer the third investigative question, “What macroeconomic outlook for Finland can be forecast over the next five years using a model incorporating ESG financial performance?”, a five-year out-of-sample forecasts (2025Q1–2029Q4) were generated using both the ESG-Only and ESG +

Global Control models. Both forecasts depict a scenario of gradual macroeconomic recovery. The ESG-Only model projects real GDP to rise from 57 142 million EUR in 2025Q1 to 58 604 million EUR by 2029Q4, while the ESG + Control model forecasts a slightly higher endpoint of 58 799 million EUR, indicating stable but modest economic growth. These represent compound annual growth rates of around 0,5 % and 0,6 %, falling short of Kokkinen’s and Jalasjoki’s forecasts for Statistics Finland of 0,7 %-0,8 % CAGR for 2020-2029 (Kokkinen et al. 2025.).

Inflation is expected to rise steadily in both models. The ESG-Only model predicts a modest increase from 129.49 to 138.24, while the ESG + Global Control model ends slightly higher at 138.89, signifying CAGR of around 1,3 %-1,4 %. Employment forecasts diverge more noticeably. The ESG-Only model projects an increase from 76.80 % to 79.05 %, whereas the ESG + Control model anticipates a more conservative rise from 76.46 % to 78.11 %. These differences reinforce earlier findings that when global ESG dynamics are accounted for, the domestic ESG index's predictive power for employment weakens considerably.

However, the models’ backtested performance reveals important limitations. All models underestimated GDP and overestimated both inflation and employment in historical evaluations. This suggests structural forecasting biases: conservative GDP projections and optimistic employment expectations. These biases may stem from the financial variables’ inability to fully account for structural challenges in the Finnish economy, such as demographic shifts and productivity constraints (Kokkinen et al. 2025).

Overall, while ESG-based models offer a useful lens for projecting Finland’s macroeconomic trajectory, their predictive accuracy is constrained by underlying structural factors that lie beyond ESG metrics alone. The forecasts support the theoretical perspectives of Stakeholder Theory and Creating Shared Value by demonstrating that ESG performance is associated with long-term economic resilience, value creation, and responsiveness to societal needs (Freeman & McVea 2001, 1-10; Porter & Kramer 2011, 4-11). However, the presence of persistent forecast biases highlights the need to complement ESG-driven approaches with a deeper understanding of Finland’s institutional context and structural economic conditions.

4.2 Reliability and Validity of the Research

In any academic study, assessing the reliability and validity of the findings is essential to understanding the strength and credibility of the results. Reliability refers to the consistency and stability of the research process and outcomes, meaning that if the study were repeated under similar conditions, the results should be similar. Validity, on the other hand, considers whether the study has accurately measured what it intended to measure and whether the conclusions drawn are trustworthy and meaningful.

This study applies a systematic and replicable research process based on established econometric methods. Tools such as the Augmented Dickey-Fuller (ADF) test for checking data stability, the Johansen cointegration test for identifying long-term relationships, and the Vector Error Correction Model (VECM) for analyzing both short- and long-term connections between variables, are commonly used in time series analysis and provide a solid methodological foundation. Furthermore, all data were collected from recognized and credible sources, including Statistics Finland, Nasdaq, and MSCI. These sources ensure that the data used are accurate, transparent, and publicly available, enhancing the overall reliability of the research.

To promote consistent measurement over time, both domestic financial index data were converted into real terms using Finland's Consumer Price Index (CPI), and all series were rebased to a common reference point (2011 = 100). This allows for fair comparison between variables and reduces the risk of distortion caused by inflation or differences in scale.

However, there are also some challenges that may affect the reliability of the results. First, the dataset covers a relatively short time period, from the first quarter of 2011 to the fourth quarter of 2024, providing only 56 quarterly observations per variable. While this is a reasonable length for macroeconomic studies, it limits the ability of the model to detect longer-term patterns or structural changes. Moreover, economic events such as the COVID-19 pandemic and Russia's invasion of Ukraine may have introduced irregularities or "shocks" into the data. These events were acknowledged in the limitations section but not directly accounted for in the model, which could impact the reliability of forecasts and interpretations.

Another potential limitation is the relatively small sample size compared to the number of variables and parameters in the VECM. Although lag selection was carefully handled to avoid overfitting, using only one lag may have missed some important short-term dynamics, while adding more lags quickly reduced model performance and introduced entropy. These trade-offs reflect a common challenge in time-series modeling with limited data.

Construct validity, which refers to how well the variables and concepts used in the study reflect the theoretical ideas behind them (Fabrigar, Wegener & Petty 2020, 320), is fairly strong. The key macroeconomic indicators used (GDP, inflation, and employment rate) are standard measures of national economic performance and are widely accepted in economic research. Similarly, the ESG indices selected (OMX Sustainability Finland GI and MSCI ACWI ESG Leaders) are well-recognized benchmarks for sustainability-focused financial performance. These indicators are therefore appropriate for answering the research question.

Internal validity, which concerns the extent to which a study can establish a trustworthy causal relationship between variables (Fabrigar et al. 2020, 319), is more limited. The VECM method is capable of identifying statistically significant relationships between variables and describing both short-term effects and long-term equilibrium conditions, but it does not prove causality. In other words, while the results may show that ESG financial performance and GDP tend to move together over time, this does not mean that ESG performance causes GDP to rise or fall. There may be other, unobserved factors influencing both. For example, general investor sentiment, government policies, or technological change could drive both economic growth and ESG performance, without one directly affecting the other.

Moreover, some relationships observed in the results, such as the negative long-run association between ESG performance and employment, raise questions about model interpretation. While this finding could reflect trends such as automation or capital-intensive business models in ESG-leading firms, it may also be an artifact of the data or model structure. Without a clearer disaggregation of the environmental, social, and governance factors or a more granular industry-specific examination, it becomes challenging to pinpoint the drivers behind the observed trends, limiting the model's interpretive clarity and warranting careful consideration of the results

External validity, referring to how well the findings of a study apply beyond the specific context in which it was conducted (Fabrigar et al. 2020, 323), is also limited. The study focuses solely on Finland, which has a unique economic and institutional environment. Finland's strong welfare state, high environmental standards, and relatively advanced ESG integration may influence how financial markets behave in relation to economic trends. As a result, the findings may not apply to countries with different social structures, regulatory frameworks, or market dynamics. In particular, developing economies or countries with less emphasis on ESG policies may experience entirely different relationships between sustainability-focused finance and macroeconomic indicators.

The inclusion of a global ESG index as a control variable aimed to account for international investor sentiment, but this index showed little influence on Finland's domestic macroeconomic outcomes, suggesting that the country's economic conditions may be somewhat insulated from global ESG movements. This may not be the case in other, more globally exposed economies.

Finally, forecast validity, or how well the models predicted future values based on past data, was mixed. The ESG-only model performed slightly better in forecasting GDP, while the ESG + Control model was more accurate in predicting inflation. The traditional financial index model provided the most accurate employment forecasts. However, all models tended to overestimate employment and underestimate GDP during the testing period. These systematic forecast errors suggest that ESG-based models can provide useful signals but should not be relied upon as sole forecasting tools. Instead, they should be viewed as complementary to traditional indicators, offering an additional layer of insight, particularly in areas related to sustainability and market sentiment.

In summary, the research process was generally reliable due to its use of consistent methods and credible data sources. However, the relatively short time period, limited sample size, and lack of structural break adjustments reduce confidence in the long-term stability of the findings. The study’s validity is strongest in its alignment of variables with theoretical concepts, but weaker in its ability to prove causality or generalize results beyond Finland. While ESG financial performance does appear to be linked to macroeconomic outcomes in Finland, these findings should be interpreted as exploratory rather than definitive. Future research could improve validity by using more detailed data, covering longer time periods, and including more countries or firm-level indicators.

4.3 Conclusions, Development, and Further Research Proposals

The study has demonstrated that ESG financial indices contain meaningful information related to Finland’s macroeconomic development. While ESG indicators were not consistently superior to traditional financial benchmarks across all dimensions, they performed competitively, particularly in forecasting GDP and inflation, and offered complementary insights into national economic trends. The findings suggest that ESG performance, particularly in a country like Finland with strong institutional alignment to sustainability and social equity, may reflect broader macroeconomic dynamics and investor expectations related to long-term value creation. These results align with both Stakeholder Theory and Creating Shared Value, which posit that companies integrating stakeholder interests and societal needs into their strategies are more likely to contribute to sustainable economic growth (Freeman & McVea 2001, 1-10; Porter & Kramer 2011, 4-11). In Finland’s case, where ESG principles are already embedded in corporate and public sector norms, financial performance aligned with ESG criteria may also serve as an early signal of structural shifts in the national economy.

Nonetheless, the empirical analysis uncovered several complexities that warrant further examination. The unexpected negative long-run relationship between ESG performance and employment, while statistically significant, raised questions about the structure of ESG-leading firms and the nature of employment growth in an advanced, innovation-driven economy. It is possible that firms excelling in ESG practices may also be those most likely to adopt automation and other capital-intensive solutions, leading to gains in output without proportional increases in labor demand. This interpretation, while tentative, suggests that strong ESG performance does not automatically translate into broad-based employment growth, and that further research is needed to disentangle these effects.

The forecast evaluation further reinforced both the promise and limitations of ESG-based models. ESG indices were able to predict GDP and inflation with comparable or slightly better accuracy than traditional indices. However, employment forecasts were less precise, and all models, regardless of specification, exhibited systematic errors such as underestimating GDP and overestimating employment. These forecast biases point to the challenges of modeling structural economic shifts using financial market data alone. They also highlight the need for continued model refinement and validation over time.

Given the exploratory nature of this thesis, several opportunities for development and further research emerge. First, a more disaggregated approach to ESG data could yield deeper insights. Instead of relying on composite ESG indices, future studies could examine the individual environmental, social, and governance components separately to determine which aspects are most closely linked to macroeconomic outcomes. This would help clarify whether, for example, environmental leadership has different predictive value than strong governance practices or social responsibility.

Second, extending the analysis beyond Finland would strengthen the external validity of the findings. Comparative studies involving other Nordic countries or members of the euro area could reveal whether the observed relationships are unique to Finland’s institutional context or more widely applicable. Likewise, incorporating firm-level data or sectoral analysis could illuminate how ESG practices differ across industries and how these differences affect aggregate economic indicators.

Third, methodological refinements could enhance the robustness of future modeling efforts. While the VECM framework was appropriate for the data and research goals of this study, more complex alternative methods may offer improved predictive performance, especially in the presence of structural breaks or nonlinearities. Including mechanisms to detect and account for major shocks, such as the COVID-19 pandemic or geopolitical disruptions, would also help ensure greater forecast stability.

Finally, future research could explore policy applications of ESG-based forecasting. As governments and central banks increasingly incorporate sustainability considerations into economic planning and financial regulation, ESG performance may evolve into a more formalized indicator of systemic risk or economic momentum. In this context, understanding the transmission mechanisms between ESG investing, corporate behavior, and macroeconomic outcomes will be essential for designing informed and resilient economic policies.

In conclusion, this thesis provides an empirical and theoretical foundation for considering ESG financial performance as a viable tool in macroeconomic forecasting, especially in sustainability-oriented economies such as Finland. While the findings should be interpreted with caution due to data limitations and modeling constraints, they offer a meaningful step toward integrating sustainable finance into the analytical toolkit of economic forecasting. As ESG data becomes more granular, standardized, and globally comparable, its potential to inform macroeconomic insight will likely continue to grow, making this an important and timely area for ongoing research and development.

4.4 Reflections on the Thesis Process and Personal Learning

The thesis process has been both intellectually rewarding and methodologically challenging. Undertaking this research gave me the opportunity to explore a topic of growing academic and practical relevance, specifically, the intersection of ESG finance and macroeconomic forecasting within the unique institutional context of Finland. Throughout the process, I gained valuable insights and experience in economic theory, financial modeling, data analysis, and academic writing. It also served as a meaningful exercise in managing a complex, independent research project while strengthening my critical thinking and problem-solving abilities.

One of the most successful aspects of the thesis was the way I was able to integrate theoretical and empirical elements into a coherent research framework. Applying Stakeholder Theory and Creating Shared Value provided a solid conceptual basis and enabled me to interpret ESG financial performance in a way that extended beyond market trends. These theories helped ground the analysis within a broader conversation about sustainability and economic development. On the empirical side, successfully implementing the Vector Error Correction Model (VECM) marked a significant personal achievement. Navigating its complexities, from identifying cointegration relationships to producing multi-period forecasts, significantly enhanced my quantitative research skills and deepened my understanding of time series econometrics.

That said, the process presented several challenges. Working with the VECM model was particularly demanding. It required a series of detailed steps, including verifying stationarity, determining the appropriate lag structure, and testing for cointegration. During the early stages, I encountered difficulties with overfitting when higher lag orders were tested, especially given the limited sample size. To address this, I prioritized model parsimony and used the Bayesian Information Criterion (BIC) to guide lag selection. Although this solution ultimately resulted in more stable models, reaching that point took considerable time and iteration.

In hindsight, there are a few things I would approach differently. For example, incorporating structural break tests or dummy variables to account for extraordinary events like the COVID-19 pandemic or Russia's invasion of Ukraine could have improved the model's robustness. These events likely introduced volatility that wasn't fully captured in the analysis. Additionally, while I relied on composite ESG indices to maintain consistency, breaking them down into their environmental, social, and governance components might have allowed for more detailed insights. Unfortunately, time and data limitations made that approach impractical within the scope of this thesis.

Despite these limitations, the experience led to substantial personal and academic development. I became more confident in using econometric software, interpreting complex statistical outputs, and visualizing economic data effectively. Beyond technical skills, I also deepened my appreciation for thoughtful research design and the importance of clear, structured communication. Perhaps most importantly, the thesis taught me perseverance. There were moments when data quality issues or unexpected modeling results were discouraging, but working through these problems taught me to be both resourceful and resilient, skills that I know will serve me well in future research or professional work.

In conclusion, the thesis journey was both demanding and fulfilling. It strengthened my ability to work independently on a complex research problem, sharpened my methodological skills, and expanded my understanding of how ESG finance can intersect with national economic trends. While there were some limitations and areas for improvement, I am proud of what I accomplished and confident that the work has laid a foundation for future academic or professional exploration in the field of sustainable finance and economic forecasting.

OpenAI's OpenAI o3 and GPT-4o language models were used in this thesis to assist in the writing process, coding, ideation and data-analysis. The produce of these models was further refined for clarity and accuracy. The application of AI was done responsibly, taking copyright issues into account. All sources cited in the report have been used correctly and are not AI-generated.

References

Algarhi, A., Karimazondo, M. 2024. The Impact of ESG on Economic Growth: Evidence from BRICS+ Countries. Theoretical Economics Letters, 14, 1484-1485. URL: https://doi.org/10.4236/tel.2024.144071. Accessed: 23 April 2025

Andersen, T., Holmström, B., Honkapohja, S., Korkman, S., Söderström, H., Vartiainen, J. 2007. THE NORDIC MODEL. The Research Institute of the Finnish Economy (ETLA), pp. 11-30. URL: https://www.etla.fi/wp-content/uploads/2012/09/B232.pdf. Accessed: 21 April 2025.

Bijelic, B., Michel, B., Mann, K., Jørgensen, A. 2025. BEHIND ESG RATINGS. OECD (Organisation for Economic Co-operation and Development). URL: https://www.oecd.org/content/dam/oecd/en/publications/reports/2025/02/behind-esg-ratings_4591b8bb/3f055f0c-en.pdf. 22 April 2025

Bilgili, F. 1998. Stationarity and cointegration tests: Comparison of Engle - Granger and Johansen methodologies Journal of Faculty of Economics and Administrative Sciences, Erciyes University No. 13, pp. 7, 10. URL: https://mpra.ub.uni-muenchen.de/75967/1/MPRA_paper_75967.pdf. 27 April 2025

Callen, T. 2019. Gross Domestic Product: An Economy's All. International Monetary Fund. URL: https://www.imf.org/en/Publications/fandd/issues/Series/Back-to-Basics/gross-domestic-product-GDP. Accessed: 22 April 2025.

Calligaris, S., Jurvanen, O., Lassi, A., Manaresi, F., Verlhac, R. The slowdown in Finnish productivity growth: causes and consequences. OECD Science, Technology and Innovation Policy Papers. URL: https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/02/the-slowdown-in-finnish-productivity-growth_6ea7cde3/c1fad5b3-en.pdf. Accessed: 22 April 2025.

Cambridge University Press. 2025. Welfare state. Cambridge Advanced Learner's Dictionary & Thesaurus. URL: https://dictionary.cambridge.org/dictionary/english/welfare-state. Accessed: 18 April 2025.

Cavanaugh, J., Neath, A. 2019. The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. WIREs Computational Statistics, pp. 1-6. URL: https://iowabiostat.github.io/research-highlights/joe/Cavanaugh_Neath_2019.pdf. Accessed: 27 April 2025.

Chen, J. 2025. What Is an Index? Examples, How It's Used, and How to Invest. Investopedia. URL: https://www.investopedia.com/Cterms/i/index.asp. Accessed: 22 April 2025.

Cheung, Y. & Lai, K. S. 1995. Lag Order and Critical Values of the Augmented Dickey-Fuller Test. Journal of business & economic statistics, 13, 3, pp. 277-280. 27 April 2025

Diab, A., Mahtani, R. 2024. ESG AUM set to top $40 trillion by 2030, anchor capital markets. Bloomberg. URL: https://www.bloomberg.com/professional/insights/sustainable-finance/esg-aum-set-to-top-40-trillion-by-2030-anchor-capital-markets/. Accessed: 22 April 2025.

Diaye, M., Ho, S. & Oueghlissi, R. 2022. ESG performance and economic growth: A panel co-integration analysis. Empirica, 49, 1, pp. 19-21. Accessed: 21 April 2025

ETLA. 2024. Etla forecasts: A turn in the economy, but Finland is set to remain the worst performer in the euro area. The Research Institute of the Finnish Economy (ETLA): URL: https://www.etla.fi/en/latest/news-and-press-releases/etla-forecasts-a-turn-in-the-economy-but-finland-is-set-to-remain-the-worst-performer-in-the-euro-area/. Accessed: 22 April 2025

ETLA. 2025. Etla forecasts: Finland’s downturn is over – EU defence budgets may save euro area growth amidst tariff turbulence. The Research Institute of the Finnish Economy (ETLA): URL: https://www.etla.fi/en/latest/etla-forecasts-finlands-downturn-is-over-eu-defence-budgets-may-save-euro-area-growth-amidst-tariff-turbulence. Accessed: 22 April 2025

Fabrigar, L. R., Wegener, D. T. & Petty, R. E. 2020-11. A Validity-Based Framework for Understanding Replication in Psychology. Personality and social psychology review, 24(4), pp. 319-323. Accessed: 4 May 2025

Finnish National Commission on Sustainable Development. 2022. Finland's strategy for sustainable development. URL: https://kestavakehitys.fi/en/finland-s-strategy-for-sustainable-development. Accessed: 23 April 2025

Freeman, R., McVea, J. 2001. A Stakeholder Approach to Strategic Management. SSRN Electronic Journal, pp. 1-10. URL: https://www.researchgate.net/publication/228320877_A_Stakeholder_Approach_to_Strategic_Management. Accessed: 15 April 2025

Friede, G., Busch, T., Bassen, A. 2015. ESG and financial performance: aggregated evidence from more than 2000 empirical studies. Journal of Sustainable Finance & Investment, 5, 4, pp. 210, 226-267. URL: https://www.tandfonline.com/doi/full/10.1080/20430795.2015.1118917#d1e141. Accessed: 18 April 2025

Giuseppe, O. 2024. ESG: From Niche to Mainstream Financial Construct. Università degli Studi di Bari “Aldo Moro” (UNIBA) - Department of Economics and Mathematical Methods, p 2. URL: http://dx.doi.org/10.2139/ssrn.4867580. Accessed: 22 April 2025

Global Sustainable Investment Alliance. 2023. Global Sustainable Investment Review 2022. URL: https://www.gsi-alliance.org/wp-content/uploads/2023/12/GSIA-Report-2022.pdf. Accessed: 22 April 2025.

Hanck, C., Arnold, M., Gerber, A., & Schmelzer, M. 2024. Introduction to Econometrics with R. University of Duisburg-Essen. Essen. E-book. Accessed: 24 April 2025.

Hodson, T. O. 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. Geoscientific Model Development, 15, 14, pp. 5482-5485. Accessed: 27 April 2025

Höflmayr, M. 2022. Inflation explained: What lies behind and what is ahead? EPRS | European Parliamentary Research Service, pp 2, 4-5. URL: https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/729352/EPRS_BRI%282022%29729352_EN.pdf. Accessed: 21 April 2025.

Iordanova, T. 2022. An Introduction to Non-Stationary Processes. Investopedia. URL: https://www.investopedia.com/articles/trading/07/stationary.asp. Accessed: 24 April 2025

Jones, T. M., Harrison, J. S. & Felps, W. 2018. How Applying Instrumental Stakeholder Theory Can Provide Sustainable Competitive Advantage. The Academy of Management review, 43(3), pp. 371-391. Accessed: 18 April 2025

Kokkinen, A., Jalasjoki P., Obstbaum M. 2025. Finnish economy's long-term growth outlook squeezed by a shrinking working-age population and weak productivity. Bank of Finland. URL: https://www.bofbulletin.fi/en/2025/articles/finnish-economy-s-long-term-growth-outlook-squeezed-by-a-shrinking-working-age-population-and-weak-productivity/. Accessed: 18 April 2025.

Koller, T., Nuttall, R. & Henisz, W. 2019. Five ways that ESG creates value. The McKinsey quarterly., pp 2-4. Accessed: 22 April 2025

Li, Q., Tang, W. & Li, Z. 2024. ESG systems and financial performance in industries with significant environmental impact: A comprehensive analysis. Frontiers in sustainability (Lausanne), 5, p. 10. Accessed: 23 April 2025

Nasdaq Baltic. 2019. About Indexes. URL: https://nasdaqbaltic.com/market-information/about-indexes/. Accessed: 28 April 2025

Parker, J. 2013. Vector Autoregression and Vector Error-Correction Models. Reed College, pp. 94-99. URL: https://www.reed.edu/economics/parker/s14/312/tschapters/S13_Ch_5.pdf. Accessed: 24 April 2025

Porter, M., Kramer, M. January/February 2011. Creating Shared Value. Harvard Business Review, pp. 4-11. URL: https://www.communitylivingbc.ca/wp-content/uploads/2018/05/Creating-Shared-Value.pdf. Accessed: 16 April 2025

Qureshi, F., Qureshi, S., Ismail, I. & Yarovaya, L. 2025. Unlocking economic insights: ESG integration, market dynamics and sustainable transitions. Energy economics, 145, p. 33. 21 April 2025. Accessed: 21 April 2025

Saha, A. K. & Khan, I. 2024. Sustainable prosperity: Unravelling the Nordic nexus of ESG, financial performance, and corporate governance. European business review, 36, 6, pp. 809-810. Accessed: 23 April 2025

Shared Value Initiative Hong Kong. 2024. What is Creating Shared Value? URL: https://sharedvaluehk.org/what-is-creating-shared-value/. Accessed: 10 May 2025

Shojaie, A. & Fox, E. B. 2021. Granger Causality: A Review and Recent Advances, pp. 291-292. Accessed: 25 April 2025

Statistics Finland. 2025. Työllisyysaste. URL: https://stat.fi/indikaattori/tyollisyysaste. Accessed: 7 April 2025

The European Foundation for the Improvement of Living and Working Conditions. 2020. Employment rate. URL: https://www.eurofound.europa.eu/en/european-industrial-relations-dictionary/employment-rate. Accessed: 22 April 2025

The Pennsylvania State University Department of Statistics. 2019a. 4.1.1 - Population is Normal. STAT 500: Applied Statistics. URL: https://online.stat.psu.edu/stat500/node/510. Accessed: 28 April 2025

The Pennsylvania State University Department of Statistics. 2019b. 3.3.3 - Probabilities for Normal Random Variables (Z-scores). STAT 500: Applied Statistics. URL: https://online.stat.psu.edu/stat500/lesson/3/3.3/3.3.3. Accessed: 28 April 2025

The United State Census Bureau. 2021. Statistical Quality Standard E1: Analyzing Data. ULR: https://www.census.gov/about/policies/quality/standards/standarde1.html. Accessed: 28 April 2025

Wang, C. 2024. The Relationship between ESG Performance and Corporate Performance - Based on Stakeholder Theory. SHS web of conferences, 190, pp. 1-4. Accessed: 18 April 2025

Whelan, T., Ulrich. A., Van Holt, T., Clark, C. 2021. ESG AND FINANCIAL PERFORMANCE: Uncovering the Relationship by Aggregating Evidence from 1,000 Plus Studies Published between

2015 – 2020. New York University Stern Center for Sustainable Business & Rockefeller Asset Management, p. 10. URL: https://www.stern.nyu.edu/sites/default/files/assets/documents/NYU-RAM_ESG-Paper_2021%20Rev_0.pdf. Accessed: 23 April 2025

Wikimedia Commons. 2008. File:Stakeholder (en).svg. URL: https://commons.wikimedia.org/wiki/File:Stakeholder_%28en%29.svg. Accessed: 28 April 2025

Appendices

Appendix 1. Thesis Data Analysis Lag 1 Python Workbook

```
from google.colab import files

# Upload all necessary files
uploaded = files.upload()
```

```
 <input type="file" id="files-88907b6b-e11c-4675-9957-1e4c255fbe98" name="files[]" multiple disabled
    style="border:none" />
 <output id="result-88907b6b-e11c-4675-9957-1e4c255fbe98">
  Upload widget is only available when the cell has been executed in the
  current browser session. Please rerun this cell to enable.
  </output>
  <script>// Copyright 2017 Google LLC
```

// // Licensed under the Apache License, Version 2.0 (the “License”); // you may not use this file except in compliance with the License. // You may obtain a copy of the License at // // http://www.apache.org/licenses/LICENSE-2.0 // // Unless required by applicable law or agreed to in writing, software // distributed under the License is distributed on an “AS IS” BASIS, // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. // See the License for the specific language governing permissions and // limitations under the License.

/** * @fileoverview Helpers for google.colab Python module. */ (function(scope) { function span(text, styleAttributes = {}) { const element = document.createElement(‘span’); element.textContent = text; for (const key of Object.keys(styleAttributes)) { element.style[key] = styleAttributes[key]; } return element; }

// Max number of bytes which will be uploaded at a time. const MAX_PAYLOAD_SIZE = 100 * 1024;

function _uploadFiles(inputId, outputId) { const steps = uploadFilesStep(inputId, outputId); const outputElement = document.getElementById(outputId); // Cache steps on the outputElement to make it available for the next call // to uploadFilesContinue from Python. outputElement.steps = steps;

return _uploadFilesContinue(outputId); }

// This is roughly an async generator (not supported in the browser yet), // where there are multiple asynchronous steps and the Python side is going // to poll for completion of each step. // This uses a Promise to block the python side on completion of each step, // then passes the result of the previous step as the input to the next step. function _uploadFilesContinue(outputId) { const outputElement = document.getElementById(outputId); const steps = outputElement.steps;

const next = steps.next(outputElement.lastPromiseValue); return Promise.resolve(next.value.promise).then((value) => { // Cache the last promise value to make it available to the next // step of the generator. outputElement.lastPromiseValue = value; return next.value.response; }); }

/** * Generator function which is called between each async step of the upload * process. * @param {string} inputId Element ID of the input file picker element. * @param {string} outputId Element ID of the output display. * @return {!Iterable<!Object>} Iterable of next steps. */ *function* uploadFilesStep(inputId, outputId) { const inputElement = document.getElementById(inputId); inputElement.disabled = false;

const outputElement = document.getElementById(outputId); outputElement.innerHTML = ’’;

const pickedPromise = new Promise((resolve) => { inputElement.addEventListener(‘change’, (e) => { resolve(e.target.files); }); });

const cancel = document.createElement(‘button’); inputElement.parentElement.appendChild(cancel); cancel.textContent = ‘Cancel upload’; const cancelPromise = new Promise((resolve) => { cancel.onclick = () => { resolve(null); }; });

// Wait for the user to pick the files. const files = yield { promise: Promise.race([pickedPromise, cancelPromise]), response: { action: ‘starting’, } };

cancel.remove();

// Disable the input element since further picks are not allowed. inputElement.disabled = true;

if (!files) { return { response: { action: ‘complete’, } }; }

for (const file of files) { const li = document.createElement(‘li’); li.append(span(file.name, {fontWeight: ‘bold’})); li.append(span(*(${file.type || 'n/a'}) - ${file.size} bytes, + last modified: ${ file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() : 'n/a'}* -)); const percent = span(‘0% done’); li.appendChild(percent);

```
outputElement.appendChild(li);

const fileDataPromise = new Promise((resolve) => {
  const reader = new FileReader();
```

```
  reader.onload = (e) => {
    resolve(e.target.result);
  };
  reader.readAsArrayBuffer(file);
});
// Wait for the data to be ready.
let fileData = yield {
  promise: fileDataPromise,
  response: {
    action: 'continue',
  }
};

// Use a chunked sending to avoid message size limits. See b/62115660.
let position = 0;
do {
  const length = Math.min(fileData.byteLength - position, MAX_PAYLOAD_SIZE);
  const chunk = new Uint8Array(fileData, position, length);
  position += length;

  const base64 = btoa(String.fromCharCode.apply(null, chunk));
  yield {
    response: {
      action: 'append',
      file: file.name,
      data: base64,
    },
  };

  let percentDone = fileData.byteLength === 0 ?
      100 :
      Math.round((position / fileData.byteLength) * 100);
  percent.textContent = `${percentDone}% done`;

} while (position < fileData.byteLength);

}
```

// All done. yield { response: { action: 'complete', } }; }

scope.google = scope.google || {}; scope.google.colab = scope.google.colab || {}; scope.google.colab._files = { _uploadFiles, _uploadFilesContinue, }; })(self);

```
Saving Control Variable.xlsx to Control Variable.xlsx
Saving Työllisyys.csv to Työllisyys.csv
Saving OMXSUSTAINFIGI.xlsx to OMXSUSTAINFIGI.xlsx
Saving OMXHGI.xlsx to OMXHGI.xlsx
Saving Kuluttajahintaindeksi.csv to Kuluttajahintaindeksi.csv
Saving BKT.csv to BKT.csv

# Tämä on muotoiltu koodina
```

VECM

```python
def clean_dataframe(df, quarter_column="Quarter"):
    """Keep only rows where Quarter is valid (e.g., 2011Q1)."""
    df = df.dropna(subset=[quarter_column])  # Drop rows where Quarter is NaN
    df = df[df[quarter_column].str.match(r'^\d{4}Q[1-4]$')]  # Keep only valid quarter formats
    df[quarter_column] = pd.PeriodIndex(df[quarter_column], freq='Q')
    return df

import pandas as pd

# Load Finnish ESG Index
esg = pd.read_excel("OMXSUSTAINFIGI.xlsx", sheet_name="QRT")
esg.columns = ["Quarter", "ESG_Index"]
esg = clean_dataframe(esg)

# Load Employment
employment = pd.read_csv("Työllisyys.csv", names=["Quarter", "Employment"], skiprows=1,
                         encoding='ISO-8859-1', sep=';', on_bad_lines='skip')
employment = clean_dataframe(employment)
employment["Employment"] = employment["Employment"].str.replace(",", ".").astype(float)
```

```
# Load Inflation
inflation = pd.read_csv("Kuluttajahintaindeksi.csv", names=["Quarter", "Inflation"], skiprows=1,
                        encoding='ISO-8859-1', sep=';', on_bad_lines='skip')
inflation = clean_dataframe(inflation)

# Load GDP
def load_gdp(filepath):
    df = pd.read_csv(filepath, sep=";", encoding="ISO-8859-1", usecols=[0, 1],
                     names=["Quarter", "GDP"], skiprows=1, on_bad_lines='skip')
    df = clean_dataframe(df)
    df["GDP"] = df["GDP"].str.replace("\xa0", "").str.replace(" ", "").str.replace(",", ".").astype(float)
    return df

gdp = load_gdp("BKT.csv")

# Load Global ESG
global_esg = pd.read_excel("Control Variable.xlsx", sheet_name="QRT")
global_esg.columns = ["Quarter", "Global_ESG"]
global_esg = clean_dataframe(global_esg)

# Load OMXHGI
omxhgi = pd.read_excel("OMXHGI.xlsx", sheet_name="QRT")
omxhgi.columns = ["Quarter", "OMXHGI"]
omxhgi = clean_dataframe(omxhgi)

# ---------- Merge All DataFrames ----------
df = esg.merge(gdp, on="Quarter")\
        .merge(employment, on="Quarter")\
        .merge(inflation, on="Quarter")\
        .merge(global_esg, on="Quarter")\
        .merge(omxhgi, on="Quarter")

# Set index
df.set_index("Quarter", inplace=True)

# ✅ Final check
print(df.head())
```

```
         ESG_Index      GDP  Employment  Inflation  Global_ESG    OMXHGI
Quarter
2011Q1    115.9973  53947.0       70.90      99.00  103.514327  113.8104
2011Q2    112.2770  53815.0       73.07      99.88  102.194959  112.0423
2011Q3     88.3871  53883.0       73.57     100.21   95.101558   89.8812
2011Q4     83.6944  53938.0       72.63     100.90   99.189156   84.5886
2012Q1     90.9911  53599.0       71.67     102.05  108.179127   92.0686
```

```
# Correlation matrix
correlation_matrix = df[["ESG_Index", "Inflation", "GDP", "Employment"]].corr()
print(correlation_matrix)
```

```
            ESG_Index  Inflation       GDP  Employment
ESG_Index    1.000000   0.663992  0.869544    0.769423
Inflation    0.663992   1.000000  0.667816    0.773022
GDP          0.869544   0.667816  1.000000    0.866731
Employment   0.769423   0.773022  0.866731    1.000000
```

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant

# Prepare data for VIF calculation
X = df[["ESG_Index", "Inflation", "GDP", "Employment"]].dropna()
X = add_constant(X)

# Calculate VIFs
vif_data = pd.DataFrame()
vif_data["Feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(vif_data)
```

```
      Feature          VIF
0       const  2816.423520
```

```
1    ESG_Index     4.340824
2    Inflation     2.620067
3          GDP     6.911448
4   Employment     5.557969

# --- Define DataFrames based on your setup ---
df_esg = df[['ESG_Index','GDP', 'Inflation', 'Employment']].dropna()
df_control = df[[ 'ESG_Index', 'GDP', 'Inflation', 'Employment','Global_ESG']].dropna()
df_trad = df[['OMXHGI','GDP', 'Inflation', 'Employment',]].dropna()

from statsmodels.tsa.api import VAR

print("🔍 Lag Order Selection for ESG-only Model")

model_esg = VAR(df_esg)
lag_order_results_esg = model_esg.select_order(maxlags=8)
print(lag_order_results_esg.summary())


🔍 Lag Order Selection for ESG-only Model
 VAR Order Selection (* highlights the minimums)
=================================================
      AIC         BIC         FPE         HQIC
-------------------------------------------------
0       24.81       24.97   5.961e+10       24.87
1       16.59      17.37*   1.602e+07       16.88
2       16.14       17.54   1.036e+07       16.67
3       15.70       17.72   6.932e+06       16.46
4       15.52       18.17   6.246e+06       16.52
5       15.57       18.84   7.445e+06       16.81
6       14.68       18.58   3.723e+06       16.15
7       14.42       18.94   3.934e+06       16.13
8      13.15*       18.30  1.790e+06*      15.10*
-------------------------------------------------

from statsmodels.tsa.stattools import adfuller
import pandas as pd

# ======================
# Set parameters
# ======================
variables = ['GDP', 'Inflation', 'Employment', 'ESG_Index', 'Global_ESG', 'OMXHGI']
lag_to_force_inflation = 0  # Based on our previous finding

# ======================
# Define flexible ADF test function
# ======================
def adf_test(series, title='', forced_lag=None):
    """
    Perform ADF test on a series.
    If forced_lag is set, use that lag instead of autolag='AIC'.
    """
    series = series.dropna()

    if forced_lag is None:
        result = adfuller(series, autolag='AIC', regression='c')
    else:
        result = adfuller(series, maxlag=forced_lag, regression='c')

    test_stat = result[0]
    p_value = result[1]
    return test_stat, p_value

# ======================
# Run ADF Tests
# ======================
results = []

# 1. Test at levels
print("\n===== ADF TEST AT LEVELS =====")
for col in variables:
    test_stat, p_value = adf_test(df[col], title=col)
```

```
    stationary = "☑ Stationary" if p_value < 0.05 else "✖ Non-stationary"
    print(f"{col}: Test Statistic = {test_stat:.2f}, P-value = {p_value:.4f} → {stationary}")
    results.append({'Variable': col, 'Difference': 'Level', 'Test Statistic': test_stat, 'P-Value': p_value,
'Stationary': stationary})

# 2. Test at first differences
print("\n===== ADF TEST AT FIRST DIFFERENCES =====")
for col in variables:
    # Special handling for Inflation (force lag 0)
    if col == 'Inflation':
        test_stat, p_value = adf_test(df[col].diff(), title=f"Δ{col}", forced_lag=lag_to_force_inflation)
    else:
        test_stat, p_value = adf_test(df[col].diff(), title=f"Δ{col}")

    stationary = "☑ Stationary" if p_value < 0.05 else "✖ Non-stationary"
    print(f"Δ{col}: Test Statistic = {test_stat:.2f}, P-value = {p_value:.4f} → {stationary}")
    results.append({'Variable': col, 'Difference': 'First Difference', 'Test Statistic': test_stat, 'P-Value'
: p_value, 'Stationary': stationary})

# ======================
# Final Results Table
# ======================
results_df = pd.DataFrame(results)
print("\n===== SUMMARY OF ADF TESTS =====")
print(results_df)
```

```
===== ADF TEST AT LEVELS =====
GDP: Test Statistic = -0.89, P-value = 0.7913 → ❌ Non-stationary
Inflation: Test Statistic = 1.84, P-value = 0.9984 → ❌ Non-stationary
Employment: Test Statistic = -1.43, P-value = 0.5692 → ❌ Non-stationary
ESG_Index: Test Statistic = -1.13, P-value = 0.7015 → ❌ Non-stationary
Global_ESG: Test Statistic = 1.88, P-value = 0.9985 → ❌ Non-stationary
OMXHGI: Test Statistic = -1.27, P-value = 0.6408 → ❌ Non-stationary

===== ADF TEST AT FIRST DIFFERENCES =====
ΔGDP: Test Statistic = -9.24, P-value = 0.0000 → ✅ Stationary
ΔInflation: Test Statistic = -2.95, P-value = 0.0399 → ✅ Stationary
ΔEmployment: Test Statistic = -3.05, P-value = 0.0301 → ✅ Stationary
ΔESG_Index: Test Statistic = -5.44, P-value = 0.0000 → ✅ Stationary
ΔGlobal_ESG: Test Statistic = -4.41, P-value = 0.0003 → ✅ Stationary
ΔOMXHGI: Test Statistic = -4.99, P-value = 0.0000 → ✅ Stationary

===== SUMMARY OF ADF TESTS =====
      Variable        Difference  Test Statistic       P-Value  \
0          GDP             Level       -0.889938  7.912678e-01
1    Inflation             Level        1.841511  9.984279e-01
2   Employment             Level       -1.427131  5.692024e-01
3    ESG_Index             Level       -1.133683  7.014543e-01
4   Global_ESG             Level        1.882023  9.984985e-01
5       OMXHGI             Level       -1.274362  6.408186e-01
6          GDP  First Difference       -9.240266  1.589545e-15
7    Inflation  First Difference       -2.949323  3.990570e-02
8   Employment  First Difference       -3.054768  3.008671e-02
9    ESG_Index  First Difference       -5.435994  2.844838e-06
10  Global_ESG  First Difference       -4.405820  2.898711e-04
11      OMXHGI  First Difference       -4.986252  2.362644e-05

          Stationary
0   ❌ Non-stationary
1   ❌ Non-stationary
2   ❌ Non-stationary
3   ❌ Non-stationary
4   ❌ Non-stationary
5   ❌ Non-stationary
6       ✅ Stationary
7       ✅ Stationary
8       ✅ Stationary
9       ✅ Stationary
```

```
10      ✅ Stationary
11      ✅ Stationary
```

```
from statsmodels.tsa.vector_ar.vecm import coint_johansen

def run_johansen_test(df_subset, k_ar_diff=1, det_order=0, label="Model"):
    result = coint_johansen(df_subset.dropna(), det_order, k_ar_diff)
    print(f"\n===== Johansen Cointegration Test: {label} =====")
    print("Trace Statistics:\n", result.lr1)
    print("Critical Values (90%, 95%, 99%):\n", result.cvt)

# --- Run for ESG-Only Model ---
run_johansen_test(df_esg[['GDP', 'Inflation', 'Employment', 'ESG_Index']], label="ESG Only")
```

```
===== Johansen Cointegration Test: ESG Only =====
Trace Statistics:
 [5.62549755e+01 1.82320440e+01 6.40288716e+00 4.43204325e-03]
Critical Values (90%, 95%, 99%):
 [[44.4929 47.8545 54.6815]
 [27.0669 29.7961 35.4628]
 [13.4294 15.4943 19.9349]
 [ 2.7055  3.8415  6.6349]]
```

```
from statsmodels.tsa.vector_ar.vecm import VECM

def run_vecm(df_subset, k_ar_diff, coint_rank, label):
    print(f"\n===== VECM Estimation: {label} =====")
    vecm_model = VECM(df_subset.dropna(), k_ar_diff=k_ar_diff, coint_rank=coint_rank, deterministic='co')
    vecm_res = vecm_model.fit()

    # Summary of the model
    print(vecm_res.summary())

    # Short-run dynamics
    print("\n=== Short-run coefficients (alpha) ===")
    print(vecm_res.alpha)

    # Long-run equilibrium (cointegration vector)
    print("\n=== Cointegration vector (beta) ===")
    print(vecm_res.beta)

    # Error correction term loadings
    print("\n=== Error Correction Term Loadings (alpha) ===")
    print(vecm_res.alpha)


# --- ESG-Only Model ---
run_vecm(df_esg[['ESG_Index','GDP', 'Inflation', 'Employment']], k_ar_diff=1, coint_rank=1, label="ESG Only")
```

```
===== VECM Estimation: ESG Only =====
Det. terms outside the coint. relation & lagged endog. parameters for equation ESG_Index
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -15.2523     35.437     -0.430      0.667     -84.708      54.204
L1.ESG_Index      0.2937      0.137      2.139      0.032       0.025       0.563
L1.GDP           -0.0027      0.002     -1.400      0.161      -0.006       0.001
L1.Inflation     -3.8403      1.758     -2.184      0.029      -7.287      -0.394
L1.Employment    -1.0891      1.208     -0.902      0.367      -3.456       1.278
Det. terms outside the coint. relation & lagged endog. parameters for equation GDP
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const          1344.2483   2805.763      0.479      0.632   -4154.947    6843.443
L1.ESG_Index     20.9203     10.870      1.925      0.054      -0.384      42.224
L1.GDP           -0.4155      0.150     -2.763      0.006      -0.710      -0.121
L1.Inflation    -14.4210    139.214     -0.104      0.917    -287.276     258.434
L1.Employment   127.9306     95.620      1.338      0.181     -59.482     315.343
Det. terms outside the coint. relation & lagged endog. parameters for equation Inflation
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
```

```
const           -1.4463      2.062     -0.701      0.483     -5.488      2.595
L1.ESG_Index    -0.0020      0.008     -0.249      0.803     -0.018      0.014
L1.GDP       -9.594e-06      0.000     -0.087      0.931     -0.000      0.000
L1.Inflation     0.7122      0.102      6.961      0.000      0.512      0.913
L1.Employment   -0.1568      0.070     -2.231      0.026     -0.295     -0.019
Det. terms outside the coint. relation & lagged endog. parameters for equation Employment
=================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const           21.6040      3.239      6.670      0.000     15.256     27.952
L1.ESG_Index     0.0301      0.013      2.401      0.016      0.006      0.055
L1.GDP          -0.0007      0.000     -4.259      0.000     -0.001     -0.000
L1.Inflation     0.6408      0.161      3.987      0.000      0.326      0.956
L1.Employment    0.3052      0.110      2.765      0.006      0.089      0.522
             Loading coefficients (alpha) for equation ESG_Index
==============================================================================
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1          -0.0002      0.000     -0.528      0.598     -0.001      0.001
                Loading coefficients (alpha) for equation GDP
==============================================================================
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           0.0145      0.032      0.459      0.646     -0.048      0.077
             Loading coefficients (alpha) for equation Inflation
==============================================================================
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1        -1.81e-05   2.33e-05     -0.777      0.437   -6.37e-05   2.75e-05
             Loading coefficients (alpha) for equation Employment
==============================================================================
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           0.0002   3.66e-05      6.717      0.000      0.000      0.000
          Cointegration relations for loading-coefficients-column 1
==============================================================================
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
beta.1        1.0000          0          0      0.000      1.000      1.000
beta.2        2.7142      0.456      5.950      0.000      1.820      3.608
beta.3      257.3635     85.160      3.022      0.003     90.452    424.275
beta.4    -3591.4718    499.678     -7.188      0.000   -4570.824  -2612.120
==============================================================================

=== Short-run coefficients (alpha) ===
[[-2.11288999e-04]
 [ 1.45417102e-02]
 [-1.80994325e-05]
 [ 2.45711975e-04]]

=== Cointegration vector (beta) ===
[[ 1.00000000e+00]
 [ 2.71421530e+00]
 [ 2.57363469e+02]
 [-3.59147181e+03]]

=== Error Correction Term Loadings (alpha) ===
[[-2.11288999e-04]
 [ 1.45417102e-02]
 [-1.80994325e-05]
 [ 2.45711975e-04]]

print("\n🔍 Lag Order Selection for ESG + Global ESG (Control) Model")
model_control = VAR(df_control)
lag_order_results_control = model_control.select_order(maxlags=8)
print(lag_order_results_control.summary())

🔍 Lag Order Selection for ESG + Global ESG (Control) Model
 VAR Order Selection (* highlights the minimums)
=================================================
      AIC         BIC         FPE         HQIC
-------------------------------------------------
0       30.69       30.89   2.141e+13       30.77
```

```
1       21.23       22.40    1.668e+09       21.67
2       20.86       23.00    1.195e+09       21.67
3       20.21       23.33    6.858e+08       21.39
4       19.78       23.87    5.346e+08       21.33
5       19.37       24.44    4.932e+08       21.28
6       17.98       24.02    2.189e+08       20.26
7       15.76       22.78    6.527e+07       18.42
8      10.04*      18.03*   1.486e+06*      13.06*
-------------------------------------------------

from statsmodels.tsa.vector_ar.vecm import coint_johansen

def run_johansen_test(df_subset, k_ar_diff=1, det_order=0, label="Model"):
    result = coint_johansen(df_subset.dropna(), det_order, k_ar_diff)
    print(f"\n===== Johansen Cointegration Test: {label} =====")
    print("Trace Statistics:\n", result.lr1)
    print("Critical Values (90%, 95%, 99%):\n", result.cvt)

# --- Run for ESG + Control Model ---
run_johansen_test(df_control[['ESG_Index', 'GDP', 'Inflation', 'Employment', 'Global_ESG']], label="ESG + Glo
bal Control")

===== Johansen Cointegration Test: ESG + Global Control =====
Trace Statistics:
 [80.07824235 36.13840783 18.58310107  8.61834993  0.5364905 ]
Critical Values (90%, 95%, 99%):
 [[65.8202 69.8189 77.8202]
 [44.4929 47.8545 54.6815]
 [27.0669 29.7961 35.4628]
 [13.4294 15.4943 19.9349]
 [ 2.7055  3.8415  6.6349]]

def run_vecm(df_subset, k_ar_diff, coint_rank, label):
    print(f"\n===== VECM Estimation: {label} =====")
    vecm_model = VECM(df_subset.dropna(), k_ar_diff=k_ar_diff, coint_rank=coint_rank, deterministic='co')
    vecm_res = vecm_model.fit()

    # Summary of the model
    print(vecm_res.summary())

    # Short-run dynamics
    print("\n=== Short-run coefficients (alpha) ===")
    print(vecm_res.alpha)

    # Long-run equilibrium (cointegration vector)
    print("\n=== Cointegration vector (beta) ===")
    print(vecm_res.beta)

    # Error correction term loadings
    print("\n=== Error Correction Term Loadings (alpha) ===")
    print(vecm_res.alpha)



# --- ESG + Control Model ---
run_vecm(df_control[['ESG_Index', 'GDP', 'Inflation', 'Employment', 'Global_ESG']], k_ar_diff=1, coint_rank=1
, label="ESG + Global Control")

===== VECM Estimation: ESG + Global Control =====
Det. terms outside the coint. relation & lagged endog. parameters for equation ESG_Index
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -16.5485     43.921     -0.377      0.706    -102.633      69.536
L1.ESG_Index      0.2945      0.164      1.790      0.073      -0.028       0.617
L1.GDP           -0.0028      0.002     -1.479      0.139      -0.006       0.001
L1.Inflation     -3.8958      1.843     -2.114      0.034      -7.507      -0.284
L1.Employment    -0.9938      1.169     -0.850      0.395      -3.286       1.298
L1.Global_ESG  7.646e-06      0.126   6.07e-05      1.000      -0.247       0.247
Det. terms outside the coint. relation & lagged endog. parameters for equation GDP
================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
```

```
const           3796.4268   3388.472      1.120      0.263   -2844.857    1.04e+04
L1.ESG_Index       9.4233     12.689      0.743      0.458     -15.447      34.294
L1.GDP            -0.4273      0.144     -2.975      0.003      -0.709      -0.146
L1.Inflation      30.4349    142.153      0.214      0.830    -248.180     309.050
L1.Employment    156.8680     90.213      1.739      0.082     -19.946     333.682
L1.Global_ESG     15.5345      9.718      1.598      0.110      -3.513      34.582
Det. terms outside the coint. relation & lagged endog. parameters for equation Inflation
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             -1.4645      2.559     -0.572      0.567      -6.479       3.550
L1.ESG_Index      -0.0004      0.010     -0.046      0.963      -0.019       0.018
L1.GDP         -2.049e-05      0.000     -0.189      0.850      -0.000       0.000
L1.Inflation       0.7097      0.107      6.611      0.000       0.499       0.920
L1.Employment     -0.1492      0.068     -2.190      0.029      -0.283      -0.016
L1.Global_ESG     -0.0022      0.007     -0.305      0.760      -0.017       0.012
Det. terms outside the coint. relation & lagged endog. parameters for equation Employment
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             26.5075      4.046      6.552      0.000      18.578      34.437
L1.ESG_Index       0.0160      0.015      1.055      0.291      -0.014       0.046
L1.GDP            -0.0007      0.000     -3.848      0.000      -0.001      -0.000
L1.Inflation       0.7653      0.170      4.509      0.000       0.433       1.098
L1.Employment      0.2422      0.108      2.249      0.025       0.031       0.453
L1.Global_ESG      0.0176      0.012      1.515      0.130      -0.005       0.040
Det. terms outside the coint. relation & lagged endog. parameters for equation Global_ESG
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const           -130.2655     54.176     -2.404      0.016    -236.448     -24.083
L1.ESG_Index       0.2295      0.203      1.131      0.258      -0.168       0.627
L1.GDP             0.0014      0.002      0.628      0.530      -0.003       0.006
L1.Inflation      -4.5642      2.273     -2.008      0.045      -9.019      -0.110
L1.Employment     -1.9998      1.442     -1.387      0.166      -4.827       0.827
L1.Global_ESG      0.0978      0.155      0.629      0.529      -0.207       0.402
             Loading coefficients (alpha) for equation ESG_Index
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           -0.0026      0.006     -0.453      0.650      -0.014       0.009
                Loading coefficients (alpha) for equation GDP
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1            0.5044      0.449      1.124      0.261      -0.375       1.384
             Loading coefficients (alpha) for equation Inflation
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           -0.0002      0.000     -0.635      0.525      -0.001       0.000
             Loading coefficients (alpha) for equation Employment
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1            0.0035      0.001      6.584      0.000       0.002       0.005
             Loading coefficients (alpha) for equation Global_ESG
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           -0.0182      0.007     -2.532      0.011      -0.032      -0.004
          Cointegration relations for loading-coefficients-column 1
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
beta.1         1.0000          0          0      0.000       1.000       1.000
beta.2         0.1481      0.034      4.344      0.000       0.081       0.215
beta.3         7.1082      8.756      0.812      0.417     -10.054      24.270
beta.4      -225.2682     31.137     -7.235      0.000    -286.295    -164.242
beta.5         0.0176      0.885      0.020      0.984      -1.717       1.753
==============================================================================

=== Short-run coefficients (alpha) ===
```

```
[[-2.63615910e-03]
 [ 5.04389624e-01]
 [-2.15213386e-04]
 [ 3.52636217e-03]
 [-1.81575349e-02]]

=== Cointegration vector (beta) ===
[[ 1.00000000e+00]
 [ 1.48088844e-01]
 [ 7.10822344e+00]
 [-2.25268171e+02]
 [ 1.76086752e-02]]

=== Error Correction Term Loadings (alpha) ===
[[-2.63615910e-03]
 [ 5.04389624e-01]
 [-2.15213386e-04]
 [ 3.52636217e-03]
 [-1.81575349e-02]]
```

```
print("\n🔍 Lag Order Selection for Traditional Index Model (OMXHGI)")
model_trad = VAR(df_trad)
lag_order_results_trad = model_trad.select_order(maxlags=8)
print(lag_order_results_trad.summary())
```

```
🔍 Lag Order Selection for Traditional Index Model (OMXHGI)
 VAR Order Selection (* highlights the minimums)
=================================================
      AIC         BIC         FPE         HQIC
-------------------------------------------------
0       25.12       25.27   8.109e+10       25.18
1       16.75      17.53*   1.888e+07       17.05
2       16.21       17.61   1.117e+07       16.74
3       15.74       17.76   7.212e+06       16.50
4       15.57       18.22   6.566e+06       16.57
5       15.52       18.80   7.095e+06       16.76
6       14.62       18.52   3.507e+06       16.09
7       14.32       18.84   3.539e+06       16.02
8      13.08*       18.22  1.661e+06*      15.02*
-------------------------------------------------
```

```
def run_johansen_test(df_subset, k_ar_diff=1, det_order=0, label="Model"):
    result = coint_johansen(df_subset.dropna(), det_order, k_ar_diff)
    print(f"\n===== Johansen Cointegration Test: {label} =====")
    print("Trace Statistics:\n", result.lr1)
    print("Critical Values (90%, 95%, 99%):\n", result.cvt)

# --- Run for Traditional Index Model ---
run_johansen_test(df_trad[['OMXHGI','GDP', 'Inflation', 'Employment' ]], label="Traditional (OMXHGI)")
```

```
===== Johansen Cointegration Test: Traditional (OMXHGI) =====
Trace Statistics:
 [5.70124826e+01 1.82805811e+01 6.01522060e+00 4.86444812e-02]
Critical Values (90%, 95%, 99%):
 [[44.4929 47.8545 54.6815]
 [27.0669 29.7961 35.4628]
 [13.4294 15.4943 19.9349]
 [ 2.7055  3.8415  6.6349]]
```

```
def run_vecm(df_subset, k_ar_diff, coint_rank, label):
    print(f"\n===== VECM Estimation: {label} =====")
    vecm_model = VECM(df_subset.dropna(), k_ar_diff=k_ar_diff, coint_rank=coint_rank, deterministic='co')
    vecm_res = vecm_model.fit()

    # Summary of the model
    print(vecm_res.summary())

    # Short-run dynamics
    print("\n=== Short-run coefficients (alpha) ===")
    print(vecm_res.alpha)
```

```
    # Long-run equilibrium (cointegration vector)
    print("\n=== Cointegration vector (beta) ===")
    print(vecm_res.beta)

    # Error correction term loadings
    print("\n=== Error Correction Term Loadings (alpha) ===")
    print(vecm_res.alpha)

# --- Traditional Index Model ---
run_vecm(df_trad[['OMXHGI','GDP', 'Inflation', 'Employment']], k_ar_diff=1, coint_rank=1, label="Traditional
(OMXHGI)")
```

```
===== VECM Estimation: Traditional (OMXHGI) =====
Det. terms outside the coint. relation & lagged endog. parameters for equation OMXHGI
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const           -13.4737     38.269     -0.352      0.725     -88.480      61.533
L1.OMXHGI         0.3870      0.137      2.828      0.005       0.119       0.655
L1.GDP           -0.0033      0.002     -1.663      0.096      -0.007       0.001
L1.Inflation     -3.7356      1.841     -2.030      0.042      -7.343      -0.128
L1.Employment    -1.0748      1.254     -0.857      0.391      -3.533       1.383
Det. terms outside the coint. relation & lagged endog. parameters for equation GDP
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const          1674.4877   2940.329      0.569      0.569   -4088.452    7437.427
L1.OMXHGI        18.4150     10.515      1.751      0.080      -2.195      39.024
L1.GDP           -0.4176      0.152     -2.747      0.006      -0.716      -0.120
L1.Inflation     -5.6930    141.412     -0.040      0.968    -282.855     271.469
L1.Employment   125.5167     96.359      1.303      0.193     -63.343     314.376
Det. terms outside the coint. relation & lagged endog. parameters for equation Inflation
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const            -1.4493      2.149     -0.675      0.500      -5.661       2.762
L1.OMXHGI        -0.0035      0.008     -0.452      0.651      -0.019       0.012
L1.GDP        -1.056e-06      0.000     -0.010      0.992      -0.000       0.000
L1.Inflation      0.7050      0.103      6.823      0.000       0.503       0.908
L1.Employment    -0.1554      0.070     -2.207      0.027      -0.293      -0.017
Det. terms outside the coint. relation & lagged endog. parameters for equation Employment
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const            22.5394      3.349      6.731      0.000      15.976      29.103
L1.OMXHGI         0.0300      0.012      2.508      0.012       0.007       0.054
L1.GDP           -0.0008      0.000     -4.348      0.000      -0.001      -0.000
L1.Inflation      0.6628      0.161      4.115      0.000       0.347       0.978
L1.Employment     0.2986      0.110      2.721      0.007       0.083       0.514
               Loading coefficients (alpha) for equation OMXHGI
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1           -0.0007      0.002     -0.440      0.660      -0.004       0.002
                Loading coefficients (alpha) for equation GDP
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1            0.0637      0.116      0.551      0.582      -0.163       0.290
             Loading coefficients (alpha) for equation Inflation
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1        -6.344e-05   8.45e-05     -0.751      0.453      -0.000       0.000
             Loading coefficients (alpha) for equation Employment
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ec1            0.0009      0.000      6.783      0.000       0.001       0.001
          Cointegration relations for loading-coefficients-column 1
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
```

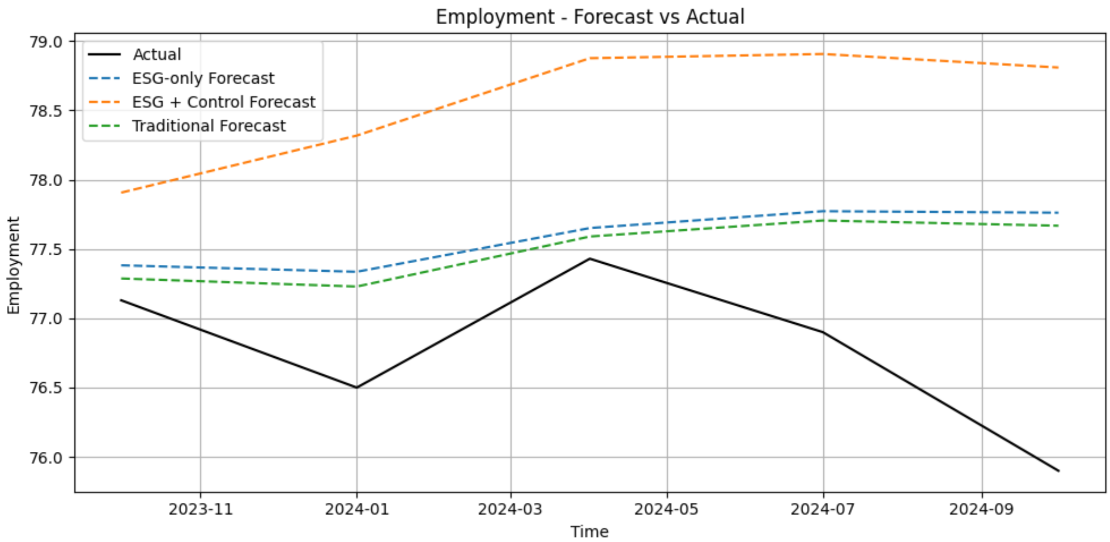
```
beta.1         1.0000          0          0      0.000       1.000       1.000
beta.2         0.7257      0.124      5.854      0.000       0.483       0.969
beta.3        73.9169     22.996      3.214      0.001      28.846     118.987
beta.4      -993.0440    135.537     -7.327      0.000   -1258.691    -727.397
==============================================================================

=== Short-run coefficients (alpha) ===
[[-6.62001577e-04]
 [ 6.37062403e-02]
 [-6.34358960e-05]
 [ 8.93341418e-04]]

=== Cointegration vector (beta) ===
[[ 1.00000000e+00]
 [ 7.25714995e-01]
 [ 7.39168960e+01]
 [-9.93043953e+02]]

=== Error Correction Term Loadings (alpha) ===
[[-6.62001577e-04]
 [ 6.37062403e-02]
 [-6.34358960e-05]
 [ 8.93341418e-04]]
```

Comparison

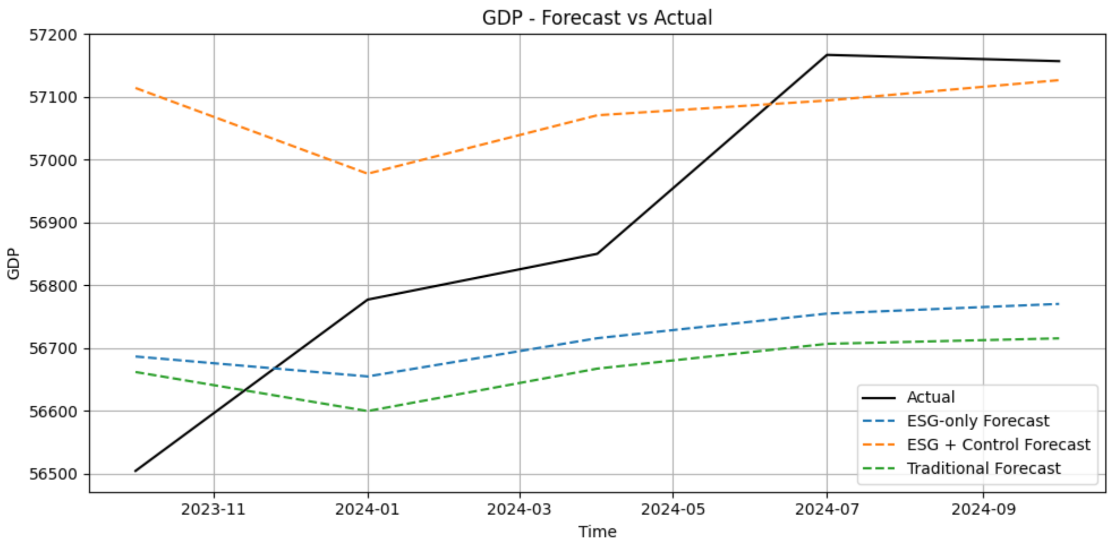
```
# --- Import Libraries ---
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.vector_ar.vecm import VECM
from sklearn.metrics import mean_squared_error, mean_absolute_error

# --- Evaluation Function ---
def forecast_metrics(y_true, y_pred):
    """Calculate RMSE and MAE."""
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mae = mean_absolute_error(y_true, y_pred)
    return {'RMSE': rmse, 'MAE': mae}

# --- VECM Forecast + Metric Evaluation ---
def vecm_forecast_metrics(df, lags, coint_rank, forecast_steps=5):
    """Fit VECM, forecast, and calculate forecast metrics."""
    df = df.dropna()
    train = df.iloc[:-forecast_steps]
    test = df.iloc[-forecast_steps:]

    model = VECM(train, k_ar_diff=lags, coint_rank=coint_rank, deterministic='co')
    res = model.fit()
    forecast = res.predict(steps=forecast_steps)

    forecast_df = pd.DataFrame(forecast, columns=df.columns, index=test.index)
    metrics = {col: forecast_metrics(test[col], forecast_df[col]) for col in df.columns}
    return metrics, forecast_df, test

# --- Run Forecast Comparisons ---
print("Running VECM Forecast Comparison...\n")

# ESG-only model
df_esg = df[['ESG_Index','GDP', 'Inflation', 'Employment', ]].dropna()
scores_esg, forecast_esg, actual_esg = vecm_forecast_metrics(df_esg, lags=1, coint_rank=1)

# ESG + Control model
df_control = df[['ESG_Index', 'GDP', 'Inflation', 'Employment', 'Global_ESG']].dropna()
scores_control, forecast_control, actual_control = vecm_forecast_metrics(df_control, lags=1, coint_rank=1)

# Traditional model (using OMXHGI)
df_trad = df[['OMXHGI','GDP', 'Inflation', 'Employment']].dropna()
scores_trad, forecast_trad, actual_trad = vecm_forecast_metrics(df_trad, lags=1, coint_rank=1)

# --- Compile Forecast Accuracy Results ---
shared_vars = set(scores_esg.keys()) & set(scores_control.keys()) & set(scores_trad.keys())
comparison = []
```

```
for var in sorted(shared_vars):
    comparison.append({
        'Variable': var,
        'ESG_RMSE': scores_esg[var]['RMSE'],
        'Control_RMSE': scores_control[var]['RMSE'],
        'Trad_RMSE': scores_trad[var]['RMSE'],
        'ESG_MAE': scores_esg[var]['MAE'],
        'Control_MAE': scores_control[var]['MAE'],
        'Trad_MAE': scores_trad[var]['MAE'],
    })

results_df = pd.DataFrame(comparison)

# --- Display Accuracy Comparison ---
print("\n=== Forecast Accuracy Comparison ===")
print(results_df.round(3))

# --- Plot Forecasts vs Actual ---
print("\n=== Forecast vs Actual Plots ===")
for var in sorted(shared_vars):
    plt.figure(figsize=(10, 5))

    # Plot actual
    plt.plot(actual_esg.index.to_timestamp(), actual_esg[var], label="Actual", color="black")

    # Plot forecasts
    if var in forecast_esg.columns:
        plt.plot(forecast_esg.index.to_timestamp(), forecast_esg[var], label="ESG-only Forecast", linestyle='
--')
    if var in forecast_control.columns:
        plt.plot(forecast_control.index.to_timestamp(), forecast_control[var], label="ESG + Control Forecast"
, linestyle='--')
    if var in forecast_trad.columns:
        plt.plot(forecast_trad.index.to_timestamp(), forecast_trad[var], label="Traditional Forecast", linest
yle='--')

    plt.title(f"{var} - Forecast vs Actual")
    plt.xlabel("Time")
    plt.ylabel(var)
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

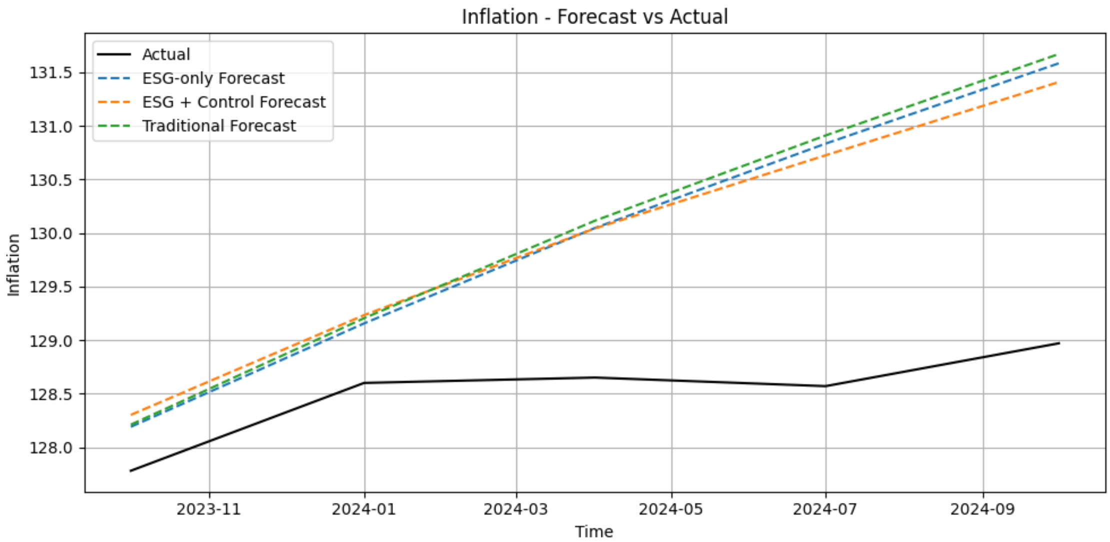
```
Running VECM Forecast Comparison...


=== Forecast Accuracy Comparison ===
     Variable  ESG_RMSE  Control_RMSE  Trad_RMSE  ESG_MAE  Control_MAE  \
0  Employment     1.004         1.923      0.933    0.809        1.792
1         GDP   277.822       305.814    315.299  247.673      226.929
2   Inflation     1.696         1.625      1.759    1.448        1.428

   Trad_MAE
0     0.724
1   284.095
2     1.508

=== Forecast vs Actual Plots ===
```


Employment - Forecast vs Actual
Actual
ESG-only Forecast
ESG + Control Forecast
Traditional Forecast
Employment
79.0
78.5
78.0
77.5
77.0
76.5
76.0
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
Time

png


GDP - Forecast vs Actual
GDP
57200
57100
57000
56900
56800
56700
56600
56500
Actual
ESG-only Forecast
ESG + Control Forecast
Traditional Forecast
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
Time

png



png

Forecast

```
import pandas as pd
import numpy as np
from statsmodels.tsa.vector_ar.vecm import VECM

# ----------- CONFIG -----------
forecast_steps = 20
lags = 1  # number of lag differences
det_order = 'co'
# ------------------------------

def vecm_manual_forecast(df, coint_rank, label="VECM"):
    """Manually forecast using fitted VECM."""
    df = df.dropna()

    # --- Fit the model ---
    vecm = VECM(df, k_ar_diff=lags, coint_rank=coint_rank, deterministic=det_order)
    res = vecm.fit()

    alpha = res.alpha
    beta = res.beta
    gamma = res.gamma  # shape: (k_endog, (k_ar_diff * k_endog))

    y = df.values.copy()
    k_endog = y.shape[1]

    # Reshape gamma into lag matrices
    gamma_lags = [gamma[:, i * k_endog:(i + 1) * k_endog] for i in range(lags)]

    # Prepare the last observations
    y_extended = list(y[-(lags + 1):])
    forecasts = []

    for _ in range(forecast_steps):
        y_last = y_extended[-1]
        y_prev = y_extended[-2]

        # Error Correction Term
        ect = beta.T @ y_last
        delta_sum = np.zeros_like(y_last)

        # Short-run dynamics
        for j in range(1, lags + 1):
            delta_y = y_extended[-j] - y_extended[-j - 1]
            delta_sum += gamma_lags[j - 1] @ delta_y
```

```
        # Forecast Δy_t
        delta_y_t = alpha @ ect + delta_sum
        y_forecast = y_last + delta_y_t

        y_extended.append(y_forecast)
        forecasts.append(y_forecast)

    # --- Create forecast DataFrame ---

    # Important fix: convert Period to Timestamp
    last_date = df.index[-1]
    if isinstance(last_date, pd.Period):
        last_date = last_date.to_timestamp()

    future_index = pd.date_range(start=last_date, periods=forecast_steps + 1, freq='QS')[1:]
    forecast_df = pd.DataFrame(forecasts, columns=df.columns, index=future_index)
    forecast_df.index.name = "Quarter"

    print(f"\n✅ Forecast ({label}) – next {forecast_steps} quarters:")
    display(forecast_df)

    return forecast_df

# ---------- Forecast ESG-Only ----------
df_esg = df[['ESG_Index','GDP', 'Inflation', 'Employment']].dropna()
forecast_esg = vecm_manual_forecast(df_esg, coint_rank=1, label="ESG-only")

# ---------- Forecast ESG + Control ----------
df_control = df[['ESG_Index','GDP', 'Inflation', 'Employment',  'Global_ESG']].dropna()
forecast_control = vecm_manual_forecast(df_control, coint_rank=1, label="ESG + Control")
```

✅ Forecast (ESG-only) – next 20 quarters:

<div>

ESG_Index

GDP

Inflation

Employment

Quarter

2025-01-01

222.412738

55797.975298

130.933858

55.193950

2025-04-01

248.815299

53859.014211

135.790314

48.497052

2025-07-01

248.845830

54394.941978

140.134670

53.564344

2025-10-01

227.006722

54636.079015

142.581509

55.430261

2026-01-01

211.451555

54080.890640

144.323667

53.325852

2026-04-01

205.519535

53584.532292

146.064189

51.929322

2026-07-01

200.624080

53415.406053

147.598437

52.004195

2026-10-01

194.424012

53318.308388

148.755998

52.102239

2027-01-01

189.139070

53167.678127

149.649419

51.793833

2027-04-01

185.527148

53023.673633

150.400417

51.486168

2027-07-01

182.744117

52927.207597

151.030111

51.355044

2027-10-01

180.273850

52858.022720

151.536959

51.280416

2028-01-01

178.189137

52796.073448

151.942804

51.184729

2028-04-01

176.550518

52741.987316

152.274148

51.092440

2028-07-01

175.247715

52699.377071

152.546079

51.026999

2028-10-01

174.172206

52666.050506

152.767305

50.979524

2029-01-01

173.285204

52638.627976

152.946599

50.938421

2029-04-01

172.566929

52615.838418

153.092455

50.902812

2029-07-01

171.986585

52597.334580

153.211391

50.874384

2029-10-01

171.513604

52582.433568

153.308231

50.852010

```
<div class="colab-df-buttons">

<button class="colab-df-convert" onclick="convertToInteractive('df-c815a82f-721d-4d35-a1dd-8ca51c8fe7a3')"
        title="Convert this dataframe to an interactive table."
        style="display:none;">



<script>
  const buttonEl =
    document.querySelector('#df-c815a82f-721d-4d35-a1dd-8ca51c8fe7a3 button.colab-df-convert');
  buttonEl.style.display =
    google.colab.kernel.accessAllowed ? 'block' : 'none';

  async function convertToInteractive(key) {
    const element = document.querySelector('#df-c815a82f-721d-4d35-a1dd-8ca51c8fe7a3');
    const dataTable =
      await google.colab.kernel.invokeFunction('convertToInteractive',
                                                [key], {});
    if (!dataTable) return;

    const docLinkHtml = 'Like what you see? Visit the ' +
      '<a target="_blank" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebo
ok</a>'
      + ' to learn more about interactive tables.';
    element.innerHTML = '';
    dataTable['output_type'] = 'display_data';
    await google.colab.output.renderOutput(dataTable, element);
    const docLink = document.createElement('div');
    docLink.innerHTML = docLinkHtml;
    element.appendChild(docLink);
  }
</script>

<div id="df-7ffe7f85-f126-462c-ade2-9c6da3aa4cb8">
  <button class="colab-df-quickchart" onclick="quickchart('df-7ffe7f85-f126-462c-ade2-9c6da3aa4cb8')"
            title="Suggest charts"
            style="display:none;">



  </button>

  <script>
    async function quickchart(key) {
      const quickchartButtonEl =
        document.querySelector('#' + key + ' button');
      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.
      quickchartButtonEl.classList.add('colab-df-spinner');
```

```
      try {
        const charts = await google.colab.kernel.invokeFunction(
            'suggestCharts', [key], {});
      } catch (error) {
        console.error('Error during call to suggestCharts:', error);
      }
      quickchartButtonEl.classList.remove('colab-df-spinner');
      quickchartButtonEl.classList.add('colab-df-quickchart-complete');
    }
    (() => {
      let quickchartButtonEl =
        document.querySelector('#df-7ffe7f85-f126-462c-ade2-9c6da3aa4cb8 button');
      quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
    })();
  </script>
</div>

</div>
```

✅ Forecast (ESG + Control) — next 20 quarters:

<div>

ESG_Index

GDP

Inflation

Employment

Global_ESG

Quarter

2025-01-01

224.314461

53624.222706

130.961143

49.954235

629.272238

2025-04-01

263.362090

52378.697546

136.459716

42.705926

723.903395

2025-07-01

265.699266

53467.769272

141.372259

46.091164

740.932602

2025-10-01

244.000586

53373.730234

144.546177

46.123347

736.961917

2026-01-01

228.640775

52643.315509

147.072884

43.968320

738.726722

2026-04-01

220.593021

52162.476417

149.382206

42.750126

742.018590

2026-07-01

213.369222

51914.714196

151.340151

42.317783

742.773456

2026-10-01

206.156582

51682.295571

152.917527

41.859297

742.613667

2027-01-01

200.221519

51450.283884

154.214776

41.334120

742.855432

2027-04-01

195.589165

51261.389602

155.302927

40.913242

743.258119

2027-07-01

191.748268

51114.125335

156.210250

40.599448

743.513384

2027-10-01

188.488123

50991.507935

156.961312

40.337360

743.672096

2028-01-01

185.775251

50887.357417

157.583974

40.111026

743.819875

2028-04-01

183.539112

50800.806921

158.101560

39.922906

743.956468

2028-07-01

181.684541

50729.490289

158.531656

39.768767

744.066924

2028-10-01

180.140594

50670.316265

158.888710

39.640934

744.155138

2029-01-01

178.857663

50621.045868

159.185142

39.534288

744.228897

2029-04-01

177.793171

50580.110938

159.431326

39.445657

744.291025

2029-07-01

176.909441

50546.148010

159.635782

39.372169

744.342549

2029-10-01

176.175374

50517.951044

159.805563

39.311167

744.385121

```
<div class="colab-df-buttons">

<button class="colab-df-convert" onclick="convertToInteractive('df-8760575a-b80d-49ac-87e2-09aaa042dfc6')"
        title="Convert this dataframe to an interactive table."
        style="display:none;">



<script>
  const buttonEl =
    document.querySelector('#df-8760575a-b80d-49ac-87e2-09aaa042dfc6 button.colab-df-convert');
  buttonEl.style.display =
    google.colab.kernel.accessAllowed ? 'block' : 'none';

  async function convertToInteractive(key) {
    const element = document.querySelector('#df-8760575a-b80d-49ac-87e2-09aaa042dfc6');
    const dataTable =
      await google.colab.kernel.invokeFunction('convertToInteractive',
                                                [key], {});
```

```
    if (!dataTable) return;

    const docLinkHtml = 'Like what you see? Visit the ' +
      '<a target="_blank" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebo
ok</a>'
      + ' to learn more about interactive tables.';
    element.innerHTML = '';
    dataTable['output_type'] = 'display_data';
    await google.colab.output.renderOutput(dataTable, element);
    const docLink = document.createElement('div');
    docLink.innerHTML = docLinkHtml;
    element.appendChild(docLink);
  }
</script>

<div id="df-34c301ba-7807-48bc-afa2-d7cc8e675688">
  <button class="colab-df-quickchart" onclick="quickchart('df-34c301ba-7807-48bc-afa2-d7cc8e675688')"
            title="Suggest charts"
            style="display:none;">



  </button>

  <script>
    async function quickchart(key) {
      const quickchartButtonEl =
        document.querySelector('#' + key + ' button');
      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.
      quickchartButtonEl.classList.add('colab-df-spinner');
      try {
        const charts = await google.colab.kernel.invokeFunction(
            'suggestCharts', [key], {});
      } catch (error) {
        console.error('Error during call to suggestCharts:', error);
      }
      quickchartButtonEl.classList.remove('colab-df-spinner');
      quickchartButtonEl.classList.add('colab-df-quickchart-complete');
    }
    (() => {
      let quickchartButtonEl =
        document.querySelector('#df-34c301ba-7807-48bc-afa2-d7cc8e675688 button');
      quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
    })();
  </script>
</div>

</div>
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaled_df = pd.DataFrame(
    scaler.fit_transform(df_esg),
    index=df_esg.index,
    columns=df_esg.columns
)

forecast_scaled = vecm_manual_forecast(scaled_df, coint_rank=1, label="ESG-only (scaled)")
```

✅ Forecast (ESG-only (scaled)) — next 20 quarters:

<div>

ESG_Index

GDP

Inflation

Employment

Quarter

2025-01-01

0.868286

0.980882

2.282347

1.105562

2025-04-01

0.802279

1.055981

2.288968

1.417828

2025-07-01

0.759387

1.028858

2.283638

1.281048

2025-10-01

0.763904

0.994000

2.287761

1.160604

2026-01-01

0.771807

0.993865

2.295516

1.186582

2026-04-01

0.766914

1.001960

2.299605

1.224937

2026-07-01

0.760447

1.000465

2.301253

1.217918

2026-10-01

0.758611

0.995694

2.303121

1.202225

2027-01-01

0.758326

0.994027

2.305247

1.200692

2027-04-01

0.757125

0.994079

2.306875

1.203971

2027-07-01

0.755644

0.993565

2.307993

1.203519

2027-10-01

0.754696

0.992627

2.308912

1.201326

2028-01-01

0.754116

0.991984

2.309727

1.200341

2028-04-01

0.753584

0.991643

2.310396

1.200224

2028-07-01

0.753081

0.991347

2.310919

1.199948

2028-10-01

0.752683

0.991046

2.311341

1.199481

2029-01-01

0.752384

0.990799

2.311690

1.199131

2029-04-01

0.752139

0.990618

2.311976

1.198929

2029-07-01

0.751932

0.990473

2.312208

1.198764

2029-10-01

0.751763

0.990349

2.312395

1.198602

```
<div class="colab-df-buttons">

<button class="colab-df-convert" onclick="convertToInteractive('df-10dc0112-0b3e-4e41-8b3c-cf33d954f5f0')"
        title="Convert this dataframe to an interactive table."
        style="display:none;">



<script>
  const buttonEl =
    document.querySelector('#df-10dc0112-0b3e-4e41-8b3c-cf33d954f5f0 button.colab-df-convert');
  buttonEl.style.display =
    google.colab.kernel.accessAllowed ? 'block' : 'none';

  async function convertToInteractive(key) {
    const element = document.querySelector('#df-10dc0112-0b3e-4e41-8b3c-cf33d954f5f0');
    const dataTable =
      await google.colab.kernel.invokeFunction('convertToInteractive',
                                                [key], {});
```

```
    if (!dataTable) return;

    const docLinkHtml = 'Like what you see? Visit the ' +
      '<a target="_blank" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebo
ok</a>'
      + ' to learn more about interactive tables.';
    element.innerHTML = '';
    dataTable['output_type'] = 'display_data';
    await google.colab.output.renderOutput(dataTable, element);
    const docLink = document.createElement('div');
    docLink.innerHTML = docLinkHtml;
    element.appendChild(docLink);
  }
</script>

<div id="df-7194e3dc-dd34-48c4-9cf1-7f8a3a7780b1">
  <button class="colab-df-quickchart" onclick="quickchart('df-7194e3dc-dd34-48c4-9cf1-7f8a3a7780b1')"
            title="Suggest charts"
            style="display:none;">



  </button>

  <script>
    async function quickchart(key) {
      const quickchartButtonEl =
        document.querySelector('#' + key + ' button');
      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.
      quickchartButtonEl.classList.add('colab-df-spinner');
      try {
        const charts = await google.colab.kernel.invokeFunction(
            'suggestCharts', [key], {});
      } catch (error) {
        console.error('Error during call to suggestCharts:', error);
      }
      quickchartButtonEl.classList.remove('colab-df-spinner');
      quickchartButtonEl.classList.add('colab-df-quickchart-complete');
    }
    (() => {
      let quickchartButtonEl =
        document.querySelector('#df-7194e3dc-dd34-48c4-9cf1-7f8a3a7780b1 button');
      quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
    })();
  </script>
</div>

</div>
```

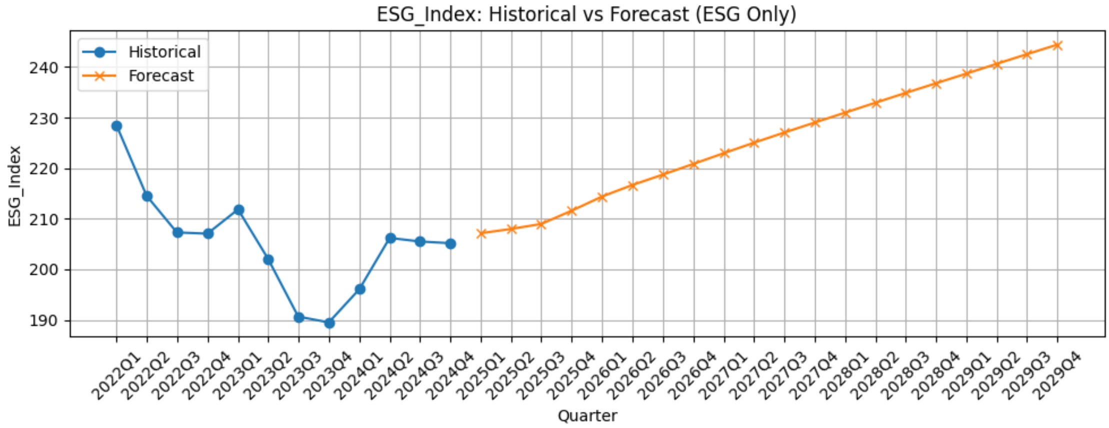
```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.vector_ar.vecm import VECM

# --- Configuration ---
forecast_steps = 20
lags = 1  # Number of lag differences
coint_rank = 1
det_order = 'co'

# --- Prepare Data ---
# Assuming 'df_esg' is your original DataFrame with ESG-only variables
scaler = StandardScaler()
scaled_df = pd.DataFrame(
    scaler.fit_transform(df_esg),
    index=df_esg.index,
    columns=df_esg.columns
)

# --- Fit VECM Model ---
model = VECM(scaled_df, k_ar_diff=lags, coint_rank=coint_rank, deterministic=det_order)
res = model.fit()
```

```python
# --- Forecast ---
forecast_scaled = res.predict(steps=forecast_steps)

# --- Inverse Transform to Original Units ---

# Fix: if index is Period, convert to Timestamp first
last_date = scaled_df.index[-1]
if isinstance(last_date, pd.Period):
    last_date = last_date.to_timestamp()

# Create future PeriodIndex, not DateIndex
future_periods = pd.period_range(start=last_date.to_period('Q'), periods=forecast_steps + 1, freq='Q')[1:]

forecast_original = pd.DataFrame(
    scaler.inverse_transform(forecast_scaled),
    index=future_periods,
    columns=scaled_df.columns
)
forecast_original.index.name = "Quarter"

# --- Print Forecast ---
print("✅ Forecast in Original Units (ESG-only) — next 20 quarters:")
print(forecast_original.round(2))
```

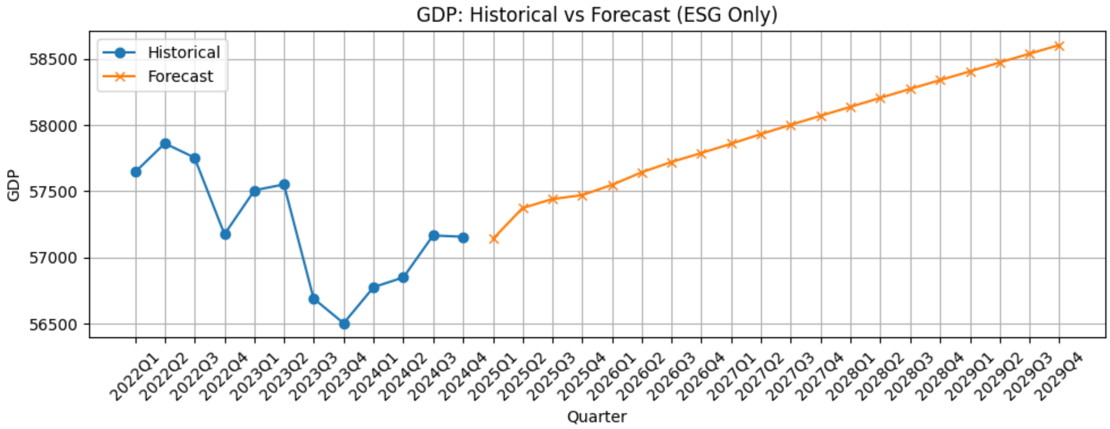
```
✅ Forecast in Original Units (ESG-only) — next 20 quarters:
         ESG_Index       GDP  Inflation  Employment
Quarter
2025Q1      207.16  57142.22     129.49       76.80
2025Q2      207.99  57373.74     129.84       77.66
2025Q3      208.98  57441.48     130.13       77.60
2025Q4      211.61  57471.66     130.51       77.46
2026Q1      214.43  57549.69     130.95       77.60
2026Q2      216.68  57643.62     131.39       77.79
2026Q3      218.71  57721.15     131.84       77.89
2026Q4      220.83  57789.24     132.29       77.96
2027Q1      222.97  57859.57     132.76       78.06
2027Q2      225.02  57931.58     133.24       78.16
2027Q3      227.02  58001.72     133.73       78.25
2027Q4      229.01  58070.02     134.22       78.34
2028Q1      230.98  58137.83     134.71       78.43
2028Q2      232.94  58205.46     135.21       78.52
2028Q3      234.87  58272.64     135.71       78.61
2028Q4      236.80  58339.35     136.21       78.70
2029Q1      238.72  58405.77     136.72       78.79
2029Q2      240.63  58471.99     137.22       78.88
2029Q3      242.54  58538.02     137.73       78.97
2029Q4      244.44  58603.89     138.24       79.05
```

```python
import matplotlib.pyplot as plt

for col in forecast_original.columns:
    plt.figure(figsize=(10, 4))

    # Prepare Historical Data
    hist = df[col].dropna().iloc[-12:]
    hist_idx = hist.index.to_timestamp()

    # Prepare Forecast Data
    forecast_idx = forecast_original.index.to_timestamp()

    # Combine indices
    combined_idx = hist_idx.append(forecast_idx)

    # Plot
    plt.plot(hist_idx, hist.values, label="Historical", marker='o')
    plt.plot(forecast_idx, forecast_original[col].values, label="Forecast", marker='x')

    plt.title(f"{col}: Historical vs Forecast (ESG Only)")
    plt.xlabel("Quarter")
    plt.ylabel(col)
    plt.legend()
    plt.grid(True)
```

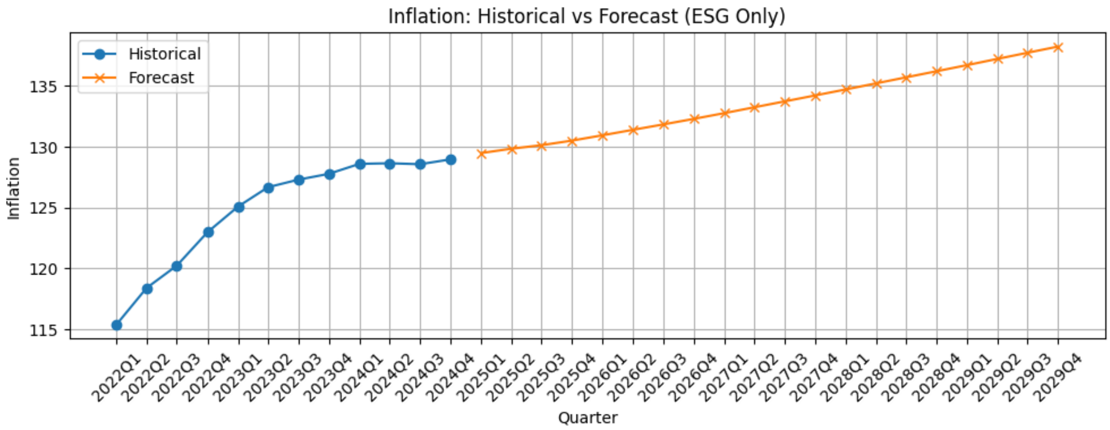
```
# 🛠 Set x-ticks only at actual points
plt.xticks(combined_idx,
           [p.to_period('Q').strftime('%YQ%q') for p in combined_idx],
           rotation=45)

plt.tight_layout()
plt.show()
```



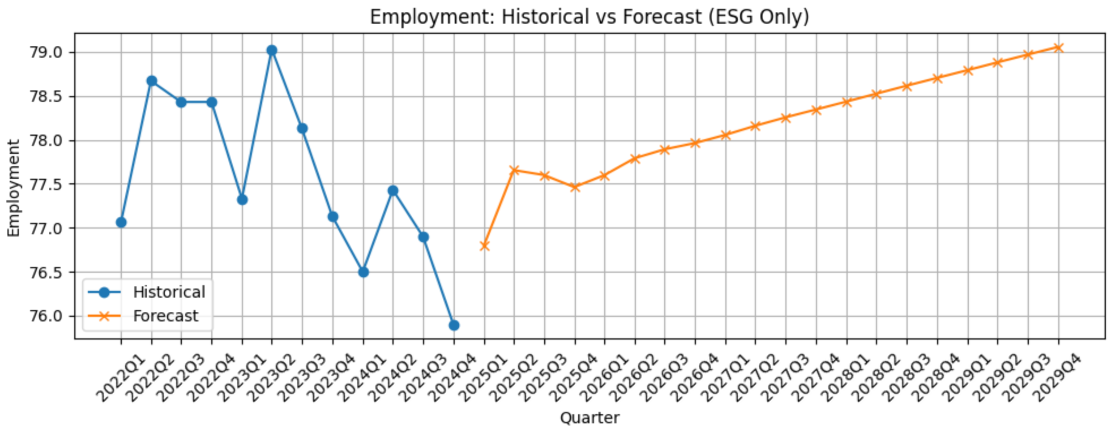
png



png



png



png

```
from sklearn.preprocessing import StandardScaler

# Scale the ESG + Control dataset
scaler = StandardScaler()
scaled_control_df = pd.DataFrame(
    scaler.fit_transform(df_control),
    index=df_control.index,
    columns=df_control.columns
)

# Forecast using the scaled data
forecast_scaled = vecm_manual_forecast(scaled_control_df, coint_rank=1, label="ESG + Control (scaled)")
```

✅ Forecast (ESG + Control (scaled)) — next 20 quarters:

<div>

ESG_Index

GDP

Inflation

Employment

Global_ESG

Quarter

2025-01-01

0.880436

1.175295

2.281643

1.037539

2.394371

2025-04-01

0.806443

1.176965

2.292881

1.100242

2.368430

2025-07-01

0.777430

1.133648

2.300931

0.995630

2.363998

2025-10-01

0.775375

1.115752

2.312280

0.959877

2.366410

2026-01-01

0.771557

1.115826

2.322265

0.970875

2.365990

2026-04-01

0.763972

1.112840

2.329421

0.968998

2.364910

2026-07-01

0.758282

1.107029

2.335156

0.956764

2.364893

2026-10-01

0.754733

1.102830

2.340177

0.948495

2.365224

2027-01-01

0.751731

1.100224

2.344413

0.944292

2.365341

2027-04-01

0.748955

1.097971

2.347872

0.940501

2.365355

2027-07-01

0.746645

1.095885

2.350725

0.936693

2.365401

2027-10-01

0.744796

1.094161

2.353107

0.933562

2.365462

2028-01-01

0.743264

1.092782

2.355090

0.931124

2.365505

2028-04-01

0.741977

1.091638

2.356735

0.929100

2.365536

2028-07-01

0.740905

1.090676

2.358099

0.927382

2.365562

2028-10-01

0.740019

1.089875

2.359232

0.925952

2.365586

2029-01-01

0.739284

1.089214

2.360174

0.924773

2.365605

2029-04-01

0.738673

1.088665

2.360956

0.923796

2.365621

2029-07-01

0.738165

1.088208

2.361605

0.922982

2.365634

2029-10-01

0.737743

1.087829

2.362144

0.922306

2.365645

```
<div class="colab-df-buttons">

<button class="colab-df-convert" onclick="convertToInteractive('df-063c1173-ff10-4e3d-8f86-1b800540bfc9')"
        title="Convert this dataframe to an interactive table."
        style="display:none;">
```

```
<script>
  const buttonEl =
    document.querySelector('#df-063c1173-ff10-4e3d-8f86-1b800540bfc9 button.colab-df-convert');
  buttonEl.style.display =
    google.colab.kernel.accessAllowed ? 'block' : 'none';

  async function convertToInteractive(key) {
    const element = document.querySelector('#df-063c1173-ff10-4e3d-8f86-1b800540bfc9');
    const dataTable =
      await google.colab.kernel.invokeFunction('convertToInteractive',
                                                [key], {});
    if (!dataTable) return;

    const docLinkHtml = 'Like what you see? Visit the ' +
      '<a target="_blank" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebo
ok</a>'
      + ' to learn more about interactive tables.';
    element.innerHTML = '';
    dataTable['output_type'] = 'display_data';
    await google.colab.output.renderOutput(dataTable, element);
    const docLink = document.createElement('div');
    docLink.innerHTML = docLinkHtml;
    element.appendChild(docLink);
  }
</script>

<div id="df-e98f433f-2ab7-4c8e-964f-463fad4729ab">
  <button class="colab-df-quickchart" onclick="quickchart('df-e98f433f-2ab7-4c8e-964f-463fad4729ab')"
          title="Suggest charts"
          style="display:none;">



  </button>

  <script>
    async function quickchart(key) {
      const quickchartButtonEl =
        document.querySelector('#' + key + ' button');
      quickchartButtonEl.disabled = true;  // To prevent multiple clicks.
      quickchartButtonEl.classList.add('colab-df-spinner');
      try {
        const charts = await google.colab.kernel.invokeFunction(
            'suggestCharts', [key], {});
      } catch (error) {
        console.error('Error during call to suggestCharts:', error);
      }
      quickchartButtonEl.classList.remove('colab-df-spinner');
      quickchartButtonEl.classList.add('colab-df-quickchart-complete');
    }
    (() => {
      let quickchartButtonEl =
        document.querySelector('#df-e98f433f-2ab7-4c8e-964f-463fad4729ab button');
      quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
    })();
  </script>
</div>

</div>
```

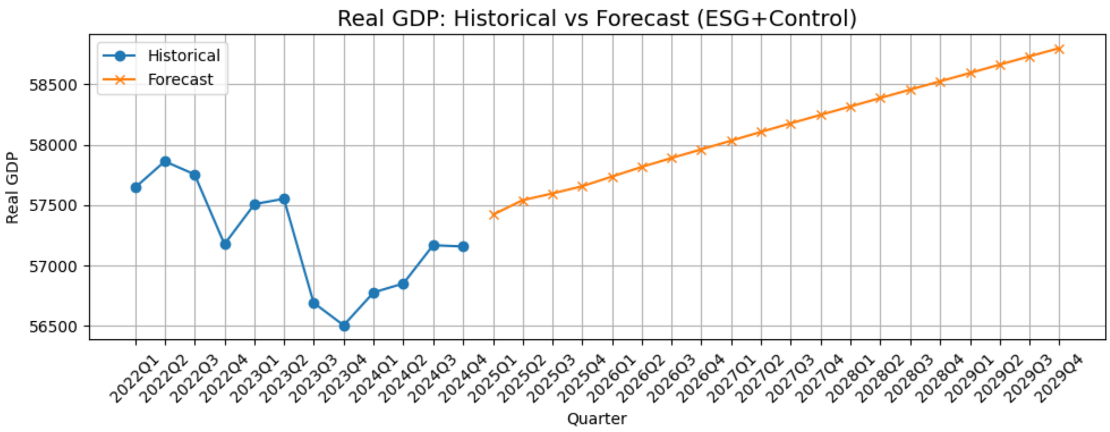
```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.vector_ar.vecm import VECM

# --- Configuration ---
forecast_steps = 20
lags = 1  # Number of lag differences
coint_rank = 1
det_order = 'co'
```

```
# --- Prepare Data ---
# Assuming 'df_control' is your DataFrame with ESG + Control variables
scaler = StandardScaler()
scaled_control_df = pd.DataFrame(
    scaler.fit_transform(df_control),
    index=df_control.index,
    columns=df_control.columns
)

# --- Fit VECM Model ---
model = VECM(scaled_control_df, k_ar_diff=lags, coint_rank=coint_rank, deterministic=det_order)
res = model.fit()

# --- Forecast ---
forecast_scaled = res.predict(steps=forecast_steps)

# --- Inverse Transform to Original Units ---

# Fix: if last index is Period, convert it safely
last_date = scaled_control_df.index[-1]
if isinstance(last_date, pd.Period):
    last_date = last_date.to_timestamp()

# Create future PeriodIndex (quarters) properly
future_periods = pd.period_range(start=last_date.to_period('Q'), periods=forecast_steps + 1, freq='Q')[1:]

forecast_original = pd.DataFrame(
    scaler.inverse_transform(forecast_scaled),
    index=future_periods,
    columns=scaled_control_df.columns
)
forecast_original.index.name = "Quarter"

# --- Print Forecast ---
print("☑ Forecast in Original Units (ESG + Control) — next 20 quarters:")
print(forecast_original.round(2))
```

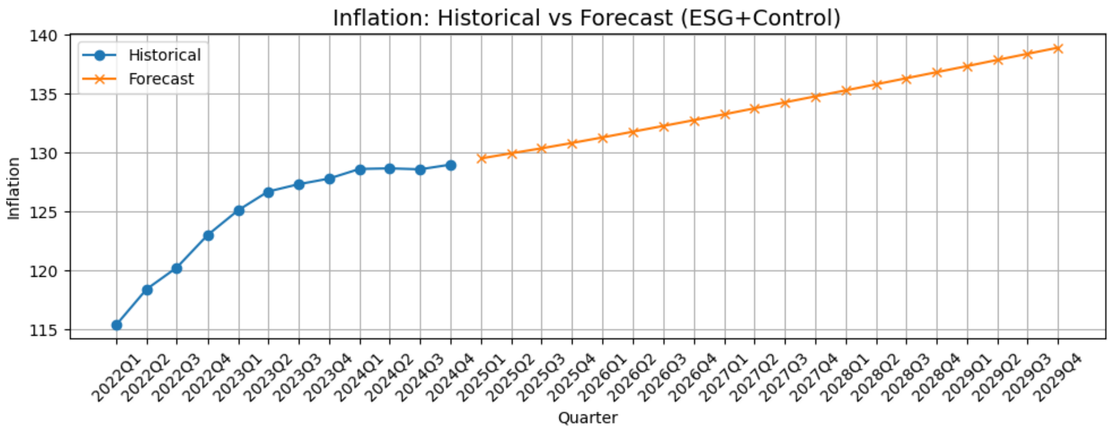
```
✅ Forecast in Original Units (ESG + Control) — next 20 quarters:
         ESG_Index       GDP  Inflation  Employment  Global_ESG
Quarter
2025Q1      207.77  57420.65     129.50       76.46      499.01
2025Q2      208.64  57540.40     129.93       76.78      504.73
2025Q3      210.14  57595.50     130.35       76.81      511.41
2025Q4      212.32  57656.13     130.81       76.85      518.75
2026Q1      214.53  57736.60     131.28       76.96      526.05
2026Q2      216.57  57815.69     131.76       77.06      533.28
2026Q3      218.57  57889.12     132.24       77.14      540.55
2026Q4      220.57  57961.34     132.74       77.22      547.86
2027Q1      222.55  58033.77     133.23       77.30      555.17
2027Q2      224.50  58105.55     133.74       77.38      562.47
2027Q3      226.44  58176.45     134.24       77.46      569.77
2027Q4      228.37  58246.79     134.75       77.53      577.07
2028Q1      230.28  58316.76     135.26       77.61      584.38
2028Q2      232.19  58386.38     135.78       77.68      591.68
2028Q3      234.08  58455.69     136.29       77.75      598.98
2028Q4      235.98  58524.75     136.81       77.83      606.29
2029Q1      237.86  58593.60     137.33       77.90      613.59
2029Q2      239.74  58662.29     137.85       77.97      620.90
2029Q3      241.62  58730.83     138.37       78.04      628.21
2029Q4      243.49  58799.26     138.89       78.11      635.51
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.vector_ar.vecm import VECM

# --- Configuration ---
forecast_steps = 20
lags = 1
coint_rank = 1
det_order = 'co'
```

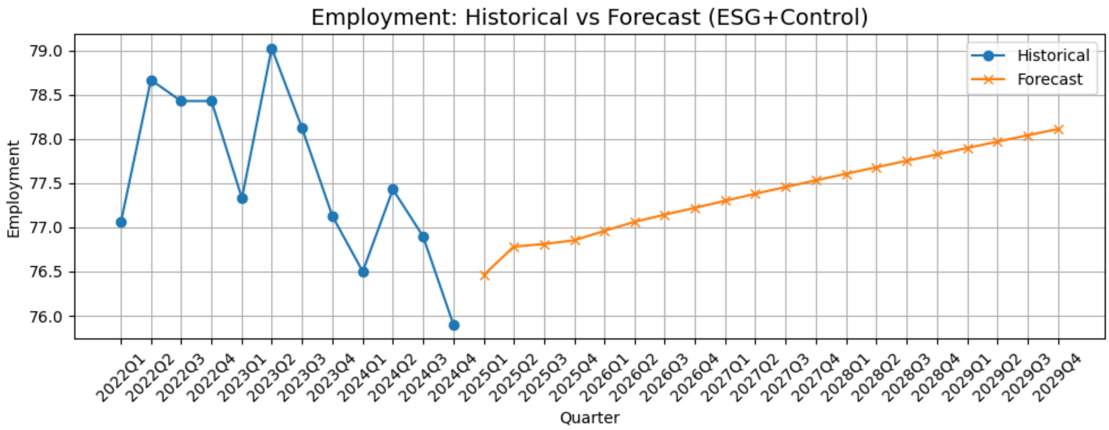
```python
# --- Prepare Data ---
scaler = StandardScaler()
scaled_control_df = pd.DataFrame(
    scaler.fit_transform(df_control),
    index=df_control.index,
    columns=df_control.columns
)

# --- Fit VECM Model ---
model = VECM(scaled_control_df, k_ar_diff=lags, coint_rank=coint_rank, deterministic=det_order)
res = model.fit()

# --- Forecast ---
forecast_scaled = res.predict(steps=forecast_steps)

# --- Inverse Transform ---
# Safely handle last date if Period
last_date = scaled_control_df.index[-1]
if isinstance(last_date, pd.Period):
    last_date = last_date.to_timestamp()

future_periods = pd.period_range(start=last_date.to_period('Q'), periods=forecast_steps + 1, freq='Q')[1:]

forecast_original = pd.DataFrame(
    scaler.inverse_transform(forecast_scaled),
    index=future_periods,
    columns=scaled_control_df.columns
)
forecast_original.index.name = "Quarter"

# --- Rename GDP to Real GDP ---
forecast_original.rename(columns={'GDP': 'Real GDP'}, inplace=True)
df_plot = df_control.rename(columns={'GDP': 'Real GDP'})

# --- Plot selected variables ---
for col in ['Real GDP', 'Inflation', 'Employment']:
    plt.figure(figsize=(10, 4))

    # Prepare Historical and Forecast data
    hist = df_plot[col].dropna().iloc[-12:]
    hist_idx = hist.index.to_timestamp()

    forecast_idx = forecast_original.index.to_timestamp()

    # Combine indices for x-axis
    combined_idx = hist_idx.append(forecast_idx)

    # Plot
    plt.plot(hist_idx, hist.values, label="Historical", marker='o')
    plt.plot(forecast_idx, forecast_original[col].values, label="Forecast", marker='x')

    plt.title(f"{col}: Historical vs Forecast (ESG+Control)", fontsize=14)
    plt.xlabel("Quarter")
    plt.ylabel(col)
    plt.legend()
    plt.grid(True)

    # 🛠 Set x-ticks only at real points, nicely labeled
    plt.xticks(combined_idx,
               [p.to_period('Q').strftime('%YQ%q') for p in combined_idx],
               rotation=45)

    plt.tight_layout()
    plt.show()
```



png



png

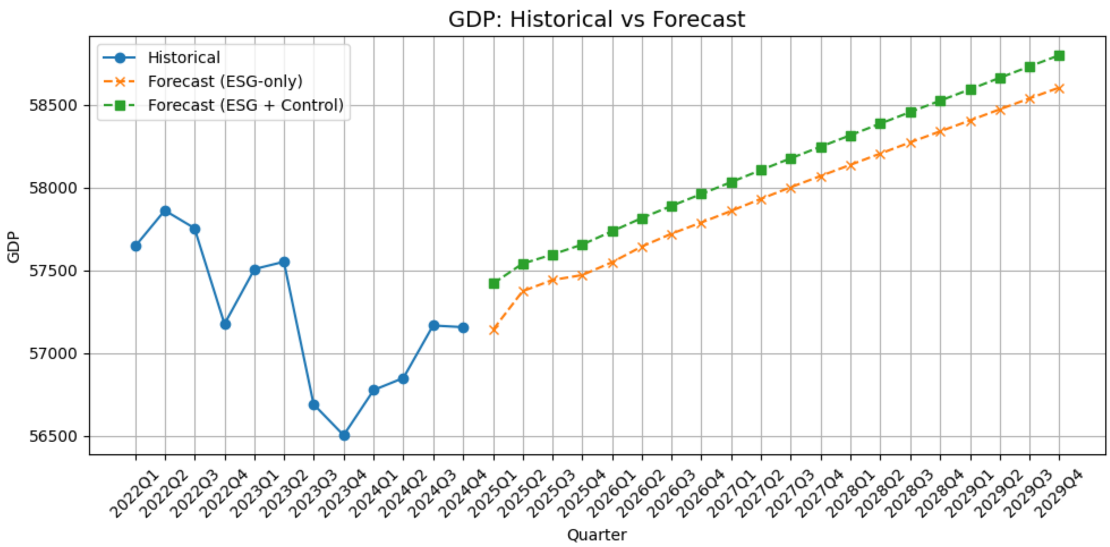


png

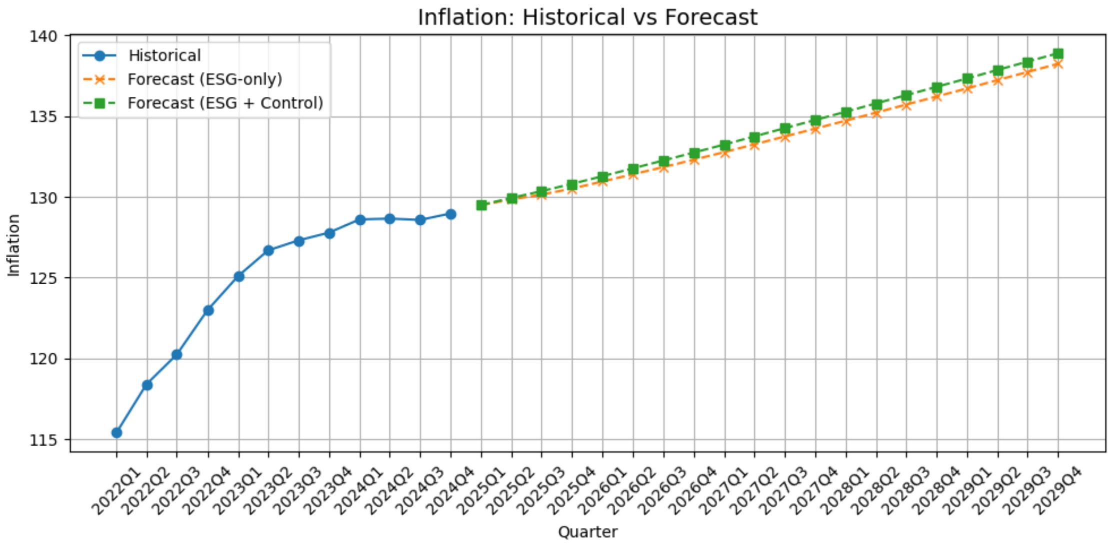
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.vector_ar.vecm import VECM

# --- Configuration ---
forecast_steps = 20
det_order = 'co'

# ---------- ESG + Control Model ----------
scaler_control = StandardScaler()
```

```
scaled_control_df = pd.DataFrame(
    scaler_control.fit_transform(df_control),
    index=df_control.index,
    columns=df_control.columns
)

model_control = VECM(scaled_control_df, k_ar_diff=1, coint_rank=1, deterministic=det_order)
res_control = model_control.fit()

forecast_scaled_control = res_control.predict(steps=forecast_steps)

# Safely handle future index creation
last_date_control = scaled_control_df.index[-1]
if isinstance(last_date_control, pd.Period):
    last_date_control = last_date_control.to_timestamp()

future_periods_control = pd.period_range(start=last_date_control.to_period('Q'), periods=forecast_steps + 1,
freq='Q')[1:]

forecast_control = pd.DataFrame(
    scaler_control.inverse_transform(forecast_scaled_control),
    index=future_periods_control,
    columns=scaled_control_df.columns
)

df_plot_control = df_control  # No renaming

# ---------- ESG-only Model ----------
scaler_esg = StandardScaler()
scaled_esg_df = pd.DataFrame(
    scaler_esg.fit_transform(df_esg),
    index=df_esg.index,
    columns=df_esg.columns
)

model_esg = VECM(scaled_esg_df, k_ar_diff=1, coint_rank=1, deterministic=det_order)
res_esg = model_esg.fit()

forecast_scaled_esg = res_esg.predict(steps=forecast_steps)

# Safely handle future index creation
last_date_esg = scaled_esg_df.index[-1]
if isinstance(last_date_esg, pd.Period):
    last_date_esg = last_date_esg.to_timestamp()

future_periods_esg = pd.period_range(start=last_date_esg.to_period('Q'), periods=forecast_steps + 1, freq='Q'
)[1:]

forecast_esg = pd.DataFrame(
    scaler_esg.inverse_transform(forecast_scaled_esg),
    index=future_periods_esg,
    columns=scaled_esg_df.columns
)

df_plot_esg = df_esg  # No renaming

# ✅ Do NOT convert indexes to strings
# ✅ Leave PeriodIndex + use to_timestamp() only when plotting

# ---------- Plotting ----------
for col in ['GDP', 'Inflation', 'Employment']:
    plt.figure(figsize=(10, 5))

    # Prepare Historical data
    historical = df_plot_esg[col].dropna().iloc[-12:]
    hist_idx = historical.index.to_timestamp()

    # Prepare Forecast data
    forecast_idx_esg = forecast_esg.index.to_timestamp()
    forecast_idx_control = forecast_control.index.to_timestamp()

    # Combine all time points
    combined_idx = hist_idx.append(forecast_idx_esg)
```

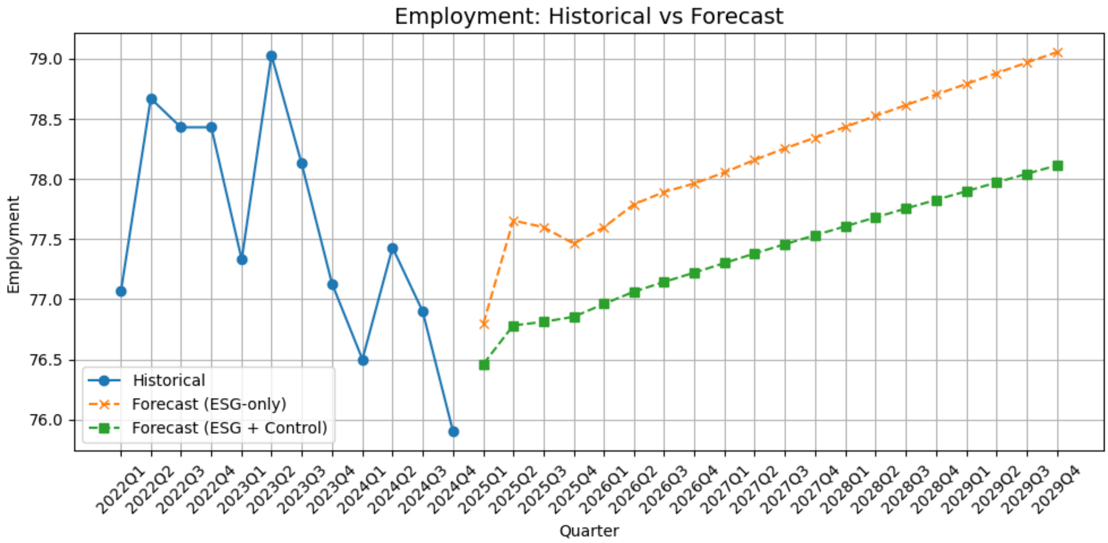
```
    # Plot historical
    plt.plot(hist_idx, historical.values, label="Historical", marker='o')

    # Plot ESG-only forecast
    plt.plot(forecast_idx_esg, forecast_esg[col].values, label="Forecast (ESG-only)", linestyle='--', marker=
'x')

    # Plot ESG + Control forecast
    plt.plot(forecast_idx_control, forecast_control[col].values, label="Forecast (ESG + Control)", linestyle=
'--', marker='s')

    plt.title(f"{col}: Historical vs Forecast", fontsize=14)
    plt.xlabel("Quarter")
    plt.ylabel(col)
    plt.grid(True)
    plt.legend()

    # 🛠 Set x-ticks nicely at all actual points
    plt.xticks(
        combined_idx,
        [p.to_period('Q').strftime('%YQ%q') for p in combined_idx],
        rotation=45
    )

    plt.tight_layout()
    plt.show()
```



png


Inflation: Historical vs Forecast
Historical
Forecast (ESG-only)
Forecast (ESG + Control)
Inflation
Quarter

png


Employment: Historical vs Forecast
Historical
Forecast (ESG-only)
Forecast (ESG + Control)
Employment
Quarter

png

Appendix 2. Thesis Data Alignment Python Workbook

```
# Raw data as a single string
data_str = """70.8 71.7 70.2 72.3 73.3 73.6 74.5 73.8 72.4 72.7 74 71.2 70.8 72.7 71.5 72.7 73.8 73.5 75.1 73.9 72.2 73 72.7 71.2 71.2 71.4 70.4 72.9 73 73.9 74.4 73.2 71.8 72.5 71.4 70.9 71.4 70.7 70.7 72.2 72.3 73.8 74.1 72.6 71.7 71.5 71.4 70.9 70.3 70.5 70.7 71.7 71.6 72.7 72.8 72.2 72.5 72 70.7 70.7 70.5 70 71.8 71.8 72.1 74.4 73.2 73 73.1 72.3 71.3 72.2 70.7 70.5 72.6 72.4 72.9 74.3 74.1 73.8 73 73.5 74.2 73.2 72.4 73.6 74 73.6 75.8 77.3 76.2 76.8 75.3 75.4 75.1 75.6 74.9 74.3 75.6 74.7 76 77.4 77.6 76.5 76.3 75.9 75.4 76.8 75.8 74.8 75.2 74.4 73.6 76.2 76.1 74.7 75.9 76 75.5 74.6 74.1 74.8 75 75 77.5 78.7 78.2 78.1 75.9 77 77 77.7 76.5 77.7 77 77.1 79.2 79.7 78.5 78.8 78 78.9 78.5 77.9 76.6 77.7 77.7 77.8 79.5 79.8 77.7 78.9 77.8 77.4 76.9 77.1 76.5 76.9 76.1 76.3 77.8 78.2 77.4 76.9 76.4 76.1 75.7 75.9"""

# Convert the string to a list of floats
monthly_values = [float(value.replace(",", ".")) for value in data_str.split()]

# Compute quarterly averages
quarters = [monthly_values[i:i+3] for i in range(0, len(monthly_values), 3)]
quarterly_averages = [round(sum(q) / len(q), 2) for q in quarters]

# Print one value per line (easy to copy into Excel)
for avg in quarterly_averages:
    print(avg)
```

```
70.9
73.07
73.57
72.63
71.67
73.33
73.73
72.3
71.0
73.27
73.13
71.6
70.93
72.77
72.8
71.27
70.5
72.0
72.5
71.13
70.77
72.77
73.1
71.93
71.27
73.2
73.63
73.63
```

73.33
75.57
76.1
75.37
74.93
76.03
76.8
76.03
75.27
74.73
75.57
75.37
74.63
77.07
77.4
77.23
77.07
78.67
78.43
78.43
77.33
79.03
78.13
77.13
76.5
77.43
76.9
75.9

```
# Raw monthly KHI data (2010=100)
monthly_values = [
    101.78, 102.40, 102.96, 103.16, 103.21, 103.51, 103.22, 103.62, 104.05, 104.31, 104.38, 104.35,
    105.01, 105.59, 105.99, 106.36, 106.37, 106.44, 106.24, 106.44, 106.86, 107.02, 106.67, 106.81,
    106.72, 107.35, 107.84, 108.00, 108.02, 107.95, 107.93, 107.76, 108.14, 108.29, 108.13, 108.53,
    108.46, 108.73, 108.98, 109.14, 108.87, 108.95, 108.83, 108.99, 109.58, 109.39, 109.18, 109.04,
    108.29, 108.57, 108.92, 108.91, 108.83, 108.82, 108.58, 108.78, 108.93, 109.10, 108.92, 108.78,
    108.32, 108.47, 108.88, 109.20, 109.16, 109.22, 109.14, 109.19, 109.38, 109.61, 109.63, 109.90,
    109.24, 109.79, 109.79, 110.12, 109.95, 110.04, 109.72, 110.00, 110.23, 110.21, 110.48, 110.44,
    110.10, 110.45, 110.67, 110.95, 111.07, 111.31, 111.22, 111.39, 111.62, 111.89, 111.87, 111.74,
    111.31, 111.84, 111.91, 112.56, 112.38, 112.45, 112.14, 112.60, 112.64, 112.74, 112.62, 112.76,
    112.41, 112.78, 112.59, 112.21, 112.18, 112.40, 112.80, 112.83, 112.82, 112.94, 112.86, 113.02,
    113.41, 113.82, 114.09, 114.52, 114.69, 114.62, 115.00, 115.28, 115.64, 116.51, 117.03, 116.95,
    118.36, 118.98, 120.70, 121.09, 122.67, 123.56, 123.96, 124.05, 125.02, 126.18, 127.72, 127.64,
    128.36, 129.44, 130.28, 130.68, 131.04, 131.30, 131.96, 131.02, 131.94, 132.31, 131.88, 132.23,
    132.56, 133.27, 133.12, 133.15, 133.00, 132.98, 133.26, 132.60, 133.01, 133.72, 133.24, 133.16
]

# Group into quarters
quarters = [monthly_values[i:i+3] for i in range(0, len(monthly_values), 3)]
quarterly_averages = [sum(q)/len(q) for q in quarters]
```

```
# Calculate average for all quarters in 2011 (1st to 4th quarter = first 4 quarters)
avg_2011 = sum(quarterly_averages[0:4]) / 4

# Rebase so 2011 = 100
rebased_quarters = [(q / avg_2011) * 100 for q in quarterly_averages]
rebased_quarters = [round(val, 2) for val in rebased_quarters]

# Output, one per line (can be pasted directly to Excel)
for q in rebased_quarters:
    print(q)
```

```
99.0
99.88
100.21
100.9
102.05
102.88
103.0
103.31
103.76
104.43
104.38
104.74
105.14
105.39
105.53
105.6
105.01
105.26
105.17
105.34
104.97
105.59
105.63
106.09
105.99
106.41
106.35
106.73
106.76
107.44
107.73
108.14
108.0
108.75
108.75
108.99
108.88
108.56
```

109.09
109.21
110.02
110.83
111.5
112.97
115.41
118.4
120.24
122.98
125.09
126.68
127.3
127.78
128.6
128.65
128.57
128.97

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
import pandas as pd

# Load your data
file_path = '/content/drive/My Drive/Thesis/OMXSUSTAINFIGI.xlsx'
df = pd.read_excel(file_path)

# Ensure the column names are clean
df.columns = [col.strip() for col in df.columns]

# Convert Trade Date to datetime
df['Trade Date'] = pd.to_datetime(df['Trade Date'])

# Create Year-Quarter labels
df['Quarter'] = df['Trade Date'].dt.to_period('Q').astype(str)

# Group by Quarter and average the Index Value
quarterly_avg = df.groupby('Quarter')['Index Value'].mean().reset_index()

# Calculate base year average (2011)
base_2011 = quarterly_avg[quarterly_avg['Quarter'].str.startswith('2011')]
base_avg = base_2011['Index Value'].mean()

# Rebase the index so 2011 average = 100
quarterly_avg['Rebased Index'] = (quarterly_avg['Index Value'] / base_avg) * 100

# Clean final DataFrame
```

```
rebased_df = quarterly_avg[['Quarter', 'Rebased Index']]
rebased_df.loc[:, 'Quarter'] = pd.PeriodIndex(rebased_df['Quarter'], freq='Q')
rebased_df = rebased_df.sort_values('Quarter').reset_index(drop=True)
rebased_df['Quarter'] = rebased_df['Quarter'].astype(str)

# Display or save
print(rebased_df)

# Optional: Save to file
# rebased_df.to_excel('/content/rebased_quarterly_index.xlsx', index=False)
```

```
   Quarter  Rebased Index
0   2011Q1     114.837330
1   2011Q2     112.142301
2   2011Q3      88.572687
3   2011Q4      84.447682
4   2012Q1      92.856405
5   2012Q2      82.886040
6   2012Q3      81.950666
7   2012Q4      88.079278
8   2013Q1      98.361346
9   2013Q2      98.785248
10  2013Q3     104.032435
11  2013Q4     118.812348
12  2014Q1     121.746057
13  2014Q2     127.731483
14  2014Q3     131.042950
15  2014Q4     131.529557
16  2015Q1     150.264312
17  2015Q2     156.513494
18  2015Q3     145.986023
19  2015Q4     151.563994
20  2016Q1     145.451826
21  2016Q2     149.093111
22  2016Q3     162.216163
23  2016Q4     162.221331
24  2017Q1     171.869304
25  2017Q2     189.975311
26  2017Q3     189.147756
27  2017Q4     187.866789
28  2018Q1     190.701397
29  2018Q2     203.077016
30  2018Q3     206.377257
31  2018Q4     191.290847
32  2019Q1     197.279401
33  2019Q2     198.805074
34  2019Q3     196.200657
35  2019Q4     201.831834
36  2020Q1     198.275917
```

```
37  2020Q2    188.113265
38  2020Q3    214.430046
39  2020Q4    226.899119
40  2021Q1    242.452341
41  2021Q2    263.326458
42  2021Q3    283.839342
43  2021Q4    279.171673
44  2022Q1    263.596636
45  2022Q2    254.020253
46  2022Q3    249.261451
47  2022Q4    254.641025
48  2023Q1    265.007093
49  2023Q2    256.042117
50  2023Q3    242.727491
51  2023Q4    242.170783
52  2024Q1    252.178127
53  2024Q2    265.296172
54  2024Q3    264.227699
55  2024Q4    264.610811
```

```python
for value in rebased_df['Rebased Index']:
    print(value)
```

```
114.83732995941476
112.142301192443
88.57268662930818
84.44768221883409
92.85640475987968
82.88603973936492
81.95066568673808
88.07927839549257
98.36134616054161
98.7852476163062
104.03243521468887
118.8123480667345
121.7460568783028
127.73148312436224
131.04294969311746
131.52955708243044
150.26431184833692
156.51349399951505
145.98602305803513
151.56399406130703
145.45182649938192
149.09311063441123
162.21616277792046
162.2213312573525
171.869304414314
189.97531069688148
```

189.14775629917574
187.86678944469196
190.7013971302995
203.0770157582046
206.37725740525684
191.2908470484886
197.2794012598904
198.80507408074521
196.20065728363244
201.83183372238918
198.27591672689698
188.1132646179521
214.43004601671714
226.8991190280072
242.45234054506005
263.3264577081325
283.83934193308585
279.1716726569836
263.59663649038293
254.02025294496272
249.26145119569213
254.64102509757504
265.0070929697798
256.04211746200434
242.7274905737149
242.17078317885233
252.17812650967826
265.29617246878854
264.2276992796857
264.6108112778363

```
from google.colab import files
uploaded = files.upload()
```

```
<input type="file" id="files-10d09804-5434-41ef-8e26-13ebeed2ed59" name="files[]" multiple disabled
   style="border:none" />
<output id="result-10d09804-5434-41ef-8e26-13ebeed2ed59">
 Upload widget is only available when the cell has been executed in the
 current browser session. Please rerun this cell to enable.
 </output>
 <script>// Copyright 2017 Google LLC
```

// // Licensed under the Apache License, Version 2.0 (the “License”); // you may not use this file except in compliance with the License. // You may obtain a copy of the License at // // http://www.apache.org/licenses/LICENSE-2.0 // // Unless required by applicable law or agreed to in writing, software // distributed under the License is distributed on an “AS IS” BASIS, // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. // See the License for the specific language governing permissions and // limitations under the License.

/** * @fileoverview Helpers for google.colab Python module. */ (function(scope) { function span(text, styleAttributes = {}) { const element = document.createElement(‘span’); element.textContent = text; for (const key of Object.keys(styleAttributes)) { element.style[key] = styleAttributes[key]; } return element; }

// Max number of bytes which will be uploaded at a time. const MAX_PAYLOAD_SIZE = 100 * 1024;

function _uploadFiles(inputId, outputId) { const steps = uploadFilesStep(inputId, outputId); const outputElement = document.getElementById(outputId); // Cache steps on the outputElement to make it available for the next call // to uploadFilesContinue from Python. outputElement.steps = steps;

return _uploadFilesContinue(outputId); }

// This is roughly an async generator (not supported in the browser yet), // where there are multiple asynchronous steps and the Python side is going // to poll for completion of each step. // This uses a Promise to block the python side on completion of each step, // then passes the result of the previous step as the input to the next step. function _uploadFilesContinue(outputId) { const outputElement = document.getElementById(outputId); const steps = outputElement.steps;

const next = steps.next(outputElement.lastPromiseValue); return Promise.resolve(next.value.promise).then((value) => { // Cache the last promise value to make it available to the next // step of the generator. outputElement.lastPromiseValue = value; return next.value.response; }); }

/** * Generator function which is called between each async step of the upload * process. * @param {string} inputId Element ID of the input file picker element. * @param {string} outputId Element ID of the output display. * @return {!Iterable<!Object>} Iterable of next steps. */ function* uploadFilesStep(inputId, outputId) { const inputElement = document.getElementById(inputId); inputElement.disabled = false;

const outputElement = document.getElementById(outputId); outputElement.innerHTML = '';

const pickedPromise = new Promise((resolve) => { inputElement.addEventListener('change', (e) => { resolve(e.target.files); }); });

const cancel = document.createElement('button'); inputElement.parentElement.appendChild(cancel); cancel.textContent = 'Cancel upload'; const cancelPromise = new Promise((resolve) => { cancel.onclick = () => { resolve(null); }; });

// Wait for the user to pick the files. const files = yield { promise: Promise.race([pickedPromise, cancelPromise]), response: { action: 'starting', } };

cancel.remove();

// Disable the input element since further picks are not allowed. inputElement.disabled = true;

if (!files) { return { response: { action: 'complete', } }; }

for (const file of files) { const li = document.createElement('li'); li.append(span(file.name, {fontWeight: 'bold'})); li.append(span(`(${file.type || 'n/a'}) - ${file.size} bytes, ` + `last modified: ${ file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() : 'n/a'} -`)); const percent = span('0% done'); li.appendChild(percent);

outputElement.appendChild(li);

```
const fileDataPromise = new Promise((resolve) => {
  const reader = new FileReader();
  reader.onload = (e) => {
    resolve(e.target.result);
  };
  reader.readAsArrayBuffer(file);
});
// Wait for the data to be ready.
let fileData = yield {
```

```
  promise: fileDataPromise,
  response: {
    action: 'continue',
  }
};

// Use a chunked sending to avoid message size limits. See b/62115660.
let position = 0;
do {
  const length = Math.min(fileData.byteLength - position, MAX_PAYLOAD_SIZE);
  const chunk = new Uint8Array(fileData, position, length);
  position += length;

  const base64 = btoa(String.fromCharCode.apply(null, chunk));
  yield {
    response: {
      action: 'append',
      file: file.name,
      data: base64,
    },
  };

  let percentDone = fileData.byteLength === 0 ?
      100 :
      Math.round((position / fileData.byteLength) * 100);
  percent.textContent = `${percentDone}% done`;

} while (position < fileData.byteLength);

}

// All done. yield { response: { action: 'complete', } }; }

scope.google = scope.google || {}; scope.google.colab = scope.google.colab || {}; scope.google.colab._files = { _uploadFiles, _uploadFilesContinue, }; })(self);
```

Saving OMXHGI.xlsx to OMXHGI.xlsx

```
import pandas as pd

# Load the manually uploaded Excel file
df = pd.read_excel('OMXHGI.xlsx')

# Ensure 'Trade Date' is a datetime
df['Trade Date'] = pd.to_datetime(df['Trade Date'])

# Add a Quarter column
df['Quarter'] = df['Trade Date'].dt.to_period('Q')

# Group by quarter and get average index value
quarterly_df = df.groupby('Quarter')['Index Value'].mean().reset_index()
```

```
# Extract only 2011 data to calculate the average index for rebasing
avg_2011 = quarterly_df[quarterly_df['Quarter'].astype(str).str.startswith('2011')]['Index Value'].mean()

# Rebase the index
quarterly_df['Rebased Index'] = quarterly_df['Index Value'] / avg_2011 * 100

# Print only the rebased index values vertically
for value in quarterly_df['Rebased Index']:
    print(value)
```

```
112.67227087317606
111.90788805238603
90.06998948238079
85.34985159205716
93.95600227575235
86.88746511018866
87.97386848012532
92.70280522336361
102.59772850289232
105.53378433950216
110.97659426861972
123.00931907365369
127.4079663360101
134.61693929795163
137.69817866270856
137.53170843326828
156.66743185277946
163.3931924142036
154.4144967480553
158.09312073890965
150.52966236660524
152.86787297660467
164.36566059068994
168.08263454175915
177.67621160008323
194.81533770038342
195.83071169820155
196.90881844156385
200.30083025843774
211.4767304192159
215.07480878118582
197.05495076842027
204.0664366025469
208.39928156278415
206.32985366273178
213.54648216288066
211.52928778238208
198.86052294119833
227.5786682903302
```

```
244.44024846394922
265.75460199536536
288.6346525471588
310.62077538941
306.2796019011394
285.6670488049222
274.02655589505997
266.7163149580788
270.803443745868
281.6970022760073
272.9411284953176
257.18436010048345
255.01771177410748
264.975129700912
277.10338447434475
274.8224555524689
273.10030878490625
```

```python
import numpy as np

# Original monthly data
monthly_data = [
    102.817, 104.693, 101.848, 101.731, 102.999, 100.685, 99.932, 92.870, 91.414, 97.155, 98.109, 101.168,
    105.450, 108.140, 109.709, 109.351, 106.644, 108.675, 113.481, 113.423, 114.141, 112.717, 114.113, 115.420,
    118.112, 123.301, 128.579, 128.999, 130.851, 127.312, 130.202, 128.065, 131.277, 136.004, 137.411, 138.186,
    135.653, 139.331, 140.421, 140.907, 146.304, 148.637, 150.388, 156.347, 157.631, 160.368, 163.912, 165.856,
    175.687, 185.683, 191.629, 187.923, 191.815, 184.283, 187.865, 172.925, 167.853, 182.673, 189.839, 181.567,
    171.919, 170.500, 175.021, 176.827, 182.472, 181.785, 188.457, 189.966, 189.181, 189.996, 197.498, 202.890,
    202.818, 211.664, 213.031, 212.570, 210.501, 208.522, 207.711, 206.313, 211.724, 219.343, 218.542, 220.568,
    224.179, 218.973, 213.867, 219.374, 226.090, 225.216, 232.184, 235.210, 236.392, 223.929, 230.456, 212.909,
    227.877, 236.359, 242.653, 251.333, 238.896, 248.402, 255.437, 253.759, 261.901, 263.040, 272.823, 277.292,
    279.303, 260.572, 227.093, 251.179, 258.620, 264.108, 262.545, 275.405, 272.980, 268.221, 291.300, 296.570,
    299.030, 305.832, 325.641, 331.055, 331.379, 346.434, 349.875, 361.434, 352.329, 376.164, 375.547, 387.149,
    371.597, 358.899, 370.977, 360.629, 353.825, 334.260, 364.369, 352.910, 327.452, 342.303, 357.837, 331.788,
    350.223, 348.128, 351.939, 350.611, 360.149, 372.089, 381.478, 379.289, 371.381, 360.704, 383.782, 396.400,
    407.636, 426.810, 442.106, 431.470, 442.950, 460.559, 460.045, 460.800, 468.895, 469.253, 500.276, 496.255,
    506.103, 500.323, 461.035
]

# Rebase to 2011 = 100 (first 12 values are 2011)
base_value = np.mean(monthly_data[:12])
rebased_data = [x / base_value * 100 for x in monthly_data]

# Convert to quarterly averages
quarterly_data = []
for i in range(0, len(rebased_data), 3):
    quarter = rebased_data[i:i+3]
    if len(quarter) == 3:  # Only take complete quarters
        quarterly_data.append(np.mean(quarter))
```

```
# Print quarterly values vertically for Excel
print("Quarterly values (2011=100):")
for value in quarterly_data:
    print(value)
```

```
Quarterly values (2011=100):
103.51432675183055
102.19495893078674
95.10155836312065
99.18915595426215
108.17912685154437
108.63787736705315
114.11711857161622
114.52032380224209
123.80307858068416
129.54833485441534
130.34537623146994
137.72587230774766
138.9987293179558
145.83916461229978
155.38157686706194
164.00448042990715
185.03907828288115
188.72715135504566
176.88931347199025
185.40045724476988
173.14067596269433
181.0521983468586
189.92605952212656
197.54847873677977
209.97221899230485
211.3374284038845
209.38163207773664
220.32505703011748
219.8452260751652
224.4163353329079
235.49393895539734
223.2833453653567
236.5322342505277
247.1534296285577
258.0168827551131
272.08991643948036
256.63527744618847
258.95713727632364
271.34540885595953
286.4567378354572
311.35574830959143
337.57747270626845
355.9040706161261
381.0741153116769
```

```
368.5640456374784
350.9103487390635
349.57759651202383
345.2935827628928
351.43769433530116
362.332266205797
378.8282119855683
381.75203547536813
427.14725607129213
446.6975232993231
465.0211097178317
490.46620395659784
491.02734517797506

# ESG and CPI values (you can also load from a file later if needed)
esg_nominal = [
    114.837330, 112.1423012, 88.57268663, 84.44768222, 92.85640476, 82.88603974,
    81.95066569, 88.0792784, 98.36134616, 98.78524762, 104.0324352, 118.8123481,
    121.7460569, 127.7314831, 131.0429497, 131.5295571, 150.2643118, 156.513494,
    145.9860231, 151.5639941, 145.4518265, 149.0931106, 162.2161628, 162.2213313,
    171.8693044, 189.9753107, 189.1477563, 187.8667894, 190.7013971, 203.0770158,
    206.3772574, 191.290847, 197.2794013, 198.8050741, 196.2006573, 201.8318337,
    198.2759167, 188.1132646, 214.430046, 226.899119, 242.4523405, 263.3264577,
    283.8393419, 279.1716727, 263.5966365, 254.0202529, 249.2614512, 254.6410251,
    265.007093, 256.0421175, 242.7274906, 242.1707832, 252.1781265, 265.2961725,
    264.2276993, 264.6108113
]

cpi = [
    99.0, 99.88, 100.21, 100.9, 102.05, 102.88, 103.0, 103.31, 103.76, 104.43,
    104.38, 104.74, 105.14, 105.39, 105.53, 105.6, 105.01, 105.26, 105.17, 105.34,
    104.97, 105.59, 105.63, 106.09, 105.99, 106.41, 106.35, 106.73, 106.76, 107.44,
    107.73, 108.14, 108.0, 108.75, 108.75, 108.99, 108.88, 108.56, 109.09, 109.21,
    110.02, 110.83, 111.5, 112.97, 115.41, 118.4, 120.24, 122.98, 125.09, 126.68,
    127.3, 127.78, 128.6, 128.65, 128.57, 128.97
]

# Check lengths match
assert len(esg_nominal) == len(cpi), "Lengths of ESG and CPI series must match!"

# Calculate real ESG index (2011 = 100)
real_esg_index = [(esg / cpi_val) * 100 for esg, cpi_val in zip(esg_nominal, cpi)]

# Display in a DataFrame
import pandas as pd

df = pd.DataFrame({
    "Nominal ESG": esg_nominal,
    "CPI (2011=100)": cpi,
    "Real ESG (2011=100)": real_esg_index
```

```
})
# Print all real ESG values
for value in real_esg_index:
    print(round(value, 4))
```

```
115.9973
112.277
88.3871
83.6944
90.9911
80.5657
79.5638
85.2573
94.797
94.5947
99.667
113.4355
115.7942
121.1989
124.176
124.5545
143.0952
148.6923
138.8096
143.8808
138.5651
141.2
153.5702
152.9092
162.1562
178.5314
177.854
176.0206
178.6263
189.0143
191.569
176.8919
182.6661
182.8093
180.4144
185.1838
182.105
173.2805
196.5625
207.764
220.3712
237.5949
254.5644
247.1202
228.4002
```

```
214.5441
207.3033
207.0589
211.8531
202.1172
190.6736
189.5217
196.095
206.2154
205.5127
205.1724

# OMXH Index (nominal) values
omxh_nominal = [
    112.6722709, 111.9078881, 90.06998948, 85.34985159, 93.95600228, 86.88746511,
    87.97386848, 92.70280522, 102.5977285, 105.5337843, 110.9765943, 123.0093191,
    127.4079663, 134.6169393, 137.6981787, 137.5317084, 156.6674319, 163.3931924,
    154.4144967, 158.0931207, 150.5296624, 152.867873, 164.3656606, 168.0826345,
    177.6762116, 194.8153377, 195.8307117, 196.9088184, 200.3008303, 211.4767304,
    215.0748088, 197.0549508, 204.0664366, 208.3992816, 206.3298537, 213.5464822,
    211.5292878, 198.8605229, 227.5786683, 244.4402485, 265.754602, 288.6346525,
    310.6207754, 306.2796019, 285.6670488, 274.0265559, 266.716315, 270.8034437,
    281.6970023, 272.9411285, 257.1843601, 255.0177118, 264.9751297, 277.1033845,
    274.8224556, 273.1003088
]

# CPI (same as before, base year 2011 = 100)
cpi = [
    99.0, 99.88, 100.21, 100.9, 102.05, 102.88, 103.0, 103.31, 103.76, 104.43,
    104.38, 104.74, 105.14, 105.39, 105.53, 105.6, 105.01, 105.26, 105.17, 105.34,
    104.97, 105.59, 105.63, 106.09, 105.99, 106.41, 106.35, 106.73, 106.76, 107.44,
    107.73, 108.14, 108.0, 108.75, 108.75, 108.99, 108.88, 108.56, 109.09, 109.21,
    110.02, 110.83, 111.5, 112.97, 115.41, 118.4, 120.24, 122.98, 125.09, 126.68,
    127.3, 127.78, 128.6, 128.65, 128.57, 128.97
]

# Check if lengths match
assert len(omxh_nominal) == len(cpi), "OMXH and CPI lengths do not match!"

# Calculate real OMXH values
real_omxh_index = [(omxh / cpi_val) * 100 for omxh, cpi_val in zip(omxh_nominal, cpi)]

# Print the real OMXH index values, rounded
for value in real_omxh_index:
    print(round(value, 4))

113.8104
112.0423
89.8812
84.5886
```

92.0686
84.4552
85.4115
89.7327
98.8798
101.057
106.3198
117.4425
121.1793
127.7322
130.4825
130.2384
149.1929
155.2282
146.8237
150.0789
143.4026
144.775
155.6051
158.434
167.6349
183.0799
184.138
184.4925
187.6179
196.8324
199.6424
182.2221
188.9504
191.6315
189.7286
195.9322
194.2775
183.1803
208.6155
223.8259
241.5512
260.4301
278.5837
271.1159
247.5237
231.4413
221.82
220.2012
225.1955
215.4572
202.0301
199.5756
206.046
215.3932

213.7532
211.7549